Springer Handbooks of Computational Statistics

Henry Horng-Shing Lu
Bernhard Schölkopf
Hongyu Zhao *Editors*

Handbook of Statistical Bioinformatics

Springer

Springer Handbooks of Computational Statistics

Series Editors

James E. Gentle
Wolfgang K. Härdle
Yuichi Mori

For further volumes:
http://www.springer.com/series/7286

Henry Horng-Shing Lu • Bernhard Schölkopf
Hongyu Zhao
Editors

Handbook of Statistical Bioinformatics

Springer

Editors

Henry Horng-Shing Lu
National Chiao Tung University
Institute of Statistics
1001 Ta Hsueh Road
30010 Hsinchu Taiwan
R.O.C.
hslu@stat.nctu.edu.tw
http://www.stat.nctu.edu.tw/∼misg/eindex.htm

Bernhard Schölkopf
MPI for Intelligent Systems
Department of Empirical Inference
Spemanstr. 38
72076 Tübingen
Germany
bernhard.schoelkopf@tiebingen.mpg
http://www.tuebingen.mpg.de/bs

Hongyu Zhao
Yale University
School of Medicine
Dept. Epidemiology and Public Health
College Street 60
New Haven, 06520 CT
USA
hongyu.zhao@yale.edu
http://bioinformatics.med.yale.edu/group/

ISBN 978-3-642-16344-9 e-ISBN 978-3-642-16345-6
DOI 10.1007/978-3-642-16345-6
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011928711

© Springer-Verlag Berlin Heidelberg 2011
This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.
The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover design: deblik, Berlin

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

Numerous fascinating breakthroughs in biotechnology have generated large volumes and diverse types of high throughput data that reveal different aspects of biological processes at the whole genome level. However, these data are highly complex and demand the development of sophisticated statistical tools, integrated with biological knowledge and implemented as computational algorithms.

This volume collects a number of statistical developments from leading researchers to survey the many active research topics in computational biology and promote the visibility of this fast evolving research area. Introductory background material can be found in books on computational statistics, such as the Springer handbook edited by Gentle et al. (2004).

The present book aims to serve as an introduction and reference on statistical methods in computational biology. It addresses students and researchers in statistics, computer science, and biological and biomedical research. We hope that most of the common topics in the field are covered in this book, and that its publication will further bridge computational statistics and computational biology to allow researchers to mine massive and diverse data sets to eventually better understand complex biological mechanisms.

The editors would like to acknowledge the encouragement and support of Wolfgang Härdle, Wen-Hsiung Li, and Wing Hung Wong for this project. We thank the authors and reviewers for their efforts and patience over the past two years. We also appreciate the web site constructed by Sebastian Stark, the latex assistance of Tung-Hung Chueh, and the general help of Springer staff, including Niels Peter Thomas, Alice Blanck and many others. Last but not the least, we acknowledge the generous support of our families while completing this challenging project. We hope that this handbook can provide a useful resource book for scholars that are interested in this exciting new area!

October 2010
Henry Horng-Shing Lu (National Chiao Tung University, Taiwan)
Bernhard Scholkopf (Max Planck Institute for Biological Cybernetics, Germany)
Hongyu Zhao (Yale University, U. S. A.)

Contents

Part I Sequence Analysis

Part II Expression Data Analysis

Part III Systems Biology

Part I
Sequence Analysis

Chapter 1
Accuracy Assessment of Consensus Sequence from Shotgun Sequencing

Lei M. Li

Abstract The significance of any genetic or biological implication based on DNA sequencing depends on its accuracy. The statistical evaluation of accuracy requires a probabilistic model of measurement error. In this chapter, we describe two statistical models of sequence assembly from shotgun sequencing respectively for the cases of haploid and diploid target genome. The first model allows us to convert quality scores into probabilities. It combines quality scores of base-calling and the power of alignment to improve sequencing accuracy. Specifically, we start with assembled contigs and represent probabilistic errors by logistic models that takes quality scores and other genomic features as covariates. Since the true sequence is unknown, an EM algorithm is used to deal with missing data. The second model describes the case in which DNA reads are from one of diploid genome, and our aim is to reconstruct the two haplotypes including phase information. The statistical model consists of sequencing errors, compositional information and haplotype memberships of each DNA fragment. Consequently, optimal haplotype sequences can be inferred by maximizing the probability among all configurations conditional on the given assembly. In the meantime, this probability together with the coverage information provides an assessment of the confidence for the reconstruction.

1.1 Introduction

Shotgun sequencing is a genome sequencing strategy. Starting with a whole genome, or a large genomic region, short random fragments are generated and sequenced using Sanger four-dye dideoxy or other techniques [1]. In Sanger sequencing each fragment is base-called from its chromatogram, i.e. a vector times series of four

L.M. Li
Institute of Systems Science, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, P.R. China
e-mail: lilei@amss.ac.cn
and
Computational Biology and Mathematics, University of Southern California, Los Angeles, CA 90089, USA

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_1,
© Springer-Verlag Berlin Heidelberg 2011

fluorescence intensities, and a sequence of A, C, G, and T, is inferred. Enough fragments should be sequenced so that almost all positions in the genome or region are covered multiple times just by chance. In shotgun sequencing projects, the sequence coverage is usually between 5 and 10. This redundancy improves the quality of the reconstructed genome. The sequence assembly in shotgun sequencing usually follows a three-step procedure: overlap-layout-consensus. That is, the base-called sequences are assembled into a contig using an ad hoc alignment algorithm that compares both strands and detects overlap between fragments. Quality values are taken into account during the alignment to eliminate low quality reads. Finally, a consensus sequence is constructed from this alignment by comparing different reads for each position. The procedure is exemplified by the *Phred/Phrap* suite of software, see [5, 6].

The accuracy of the consensus sequence depends on the coverage – namely, by how many independent observations we have for each nucleotide base-pair in the genome – and the performance of base-calling algorithm. The quality values of base-calling play a crucial role in the construction of consensus. If they are misleading or interpreted incorrectly, the consensus sequence will be less reliable. The *Phred* quality scores for base-calling are defined from sequencing traces in such a way that they have a probabilistic interpretation. This is achieved by training a model on a large amount of data. However, sample preparations and sequencing machines may work under different conditions in practice and quality scores need to be adjusted. Also the information given by quality scores is incomplete in the sense that they do not tell us error patterns. We do observe that each nucleotide base has its specific error pattern that varies across the range of quality values.

Churchill and Waterman [3] proposed another model to define a consensus. It is based on an assembly without assuming the availability of quality values. The parameters in the model include composition probabilities and sequencing error rates and are estimated by an E-M algorithm based on the alignment. The consensus is defined by the probability of the target sequence conditional on observations. This offers an evaluation of reliability of the estimated target sequence.

In the first half of this chapter, we describe an accuracy assessment method that combines quality scores of base-calling and the idea in Churchill and Waterman [3] to improve sequencing accuracy. Specifically, we start with assembled contigs and quality scores to build up complete probabilistic error models. One option is to represent the error pattern of each nucleotide by a multinomial model. Since the true sequence is unknown, we develop an EM algorithm to deal with missing data. In a more sophisticated mixture of logistic model, we take quality scores as covariates. To parsimoniously represent the nonlinear effect of quality scores, we adopt simple piecewise linear functions in the regression model. The model is trained by a procedure combining an EM algorithm, the BIC criterion and backward deletion. The training results in calibration of quality values that lead to more accurate consensus construction.

Even though the objective of shotgun sequencing has been to determine a haploid consensus sequence, fragments are from the diploid genome in an eukaryotic organism. In this diploid reconstruction problem, the origins of fragments are unknown.

It is necessary to differentiate polymorphisms from sequencing errors, and then to infer the phases between adjacent polymorphisms. The diploid shotgun sequencing problem was formulated in [10] by a graph theoretic approach. To deal with errors seen in fragments, they defined several combinatorial problems such as Minimum Fragment Removal (MFR), Minimum Snip Removal (MSR) and Longest Haplotype Reconstruction (LHR). Further development along this direction can be found in [17]. Unlike haploid problem, in which it is sufficient to provide the quality measure of each consensus base, the reconstruction of diploid genome need to consider multiple loci jointly because the phase between polymorphisms should be considered in accuracy assessment.

In the second half of this chapter, we describe a probabilistic model for the sequence assembly from a diploid genome sequencing project. Consequently, the probabilities of different haplotypes (conditional on the assembly layout) can be calculated and the optimal consensus sequences can be inferred by maximizing this probability. In the meantime, this probability together with the coverage information provides an assessment of the confidence for the reconstruction.

1.2 Adjustment of Quality Scores from Alignment and Improvement of Sequencing Accuracy

1.2.1 Sequencing Data

The first source of data in this chapter comes from the *Campylobacter jejuni* whole-genome shotgun sequencing project [20]. The raw data, generated on ABI 373 and 377 automated sequencers were downloaded from the Sanger Center (ftp.sanger.ac.uk/pub/pathogens/cj). The total length of the genome sequence is 1,641,481 bp. There are 33,824 reads and the average coverage is ten folds. The sequence assembly was obtained by *Phrap* (see http://www.phrap.org). We tested our methods on the first 100 kb of the reference sequence and the corresponding reads. The coverage of the *C. jejuni* sequencing project is unusually high, so we randomly removed some reads to decrease the average coverage from ten-fold to six-fold. Since the reference sequence was obtained on reads of ten-fold, we will assume that it is close to the true sequence later when we calculate single base discrepancy (SBD).

To test our methods on data obtained using another sequencing technology, we analyzed data from an *Arabidopsis thaliana* re-sequencing project carried out at USC (http://walnut.usc.edu/2010) using Beckman Coulter CEQ automated sequencers. These data were obtained as part of a polymorphism survey, and thus contain different haplotypes. Since we are only interested in sequencing error, we selected about 500 kb of raw data from non-polymorphic regions.

$$\begin{array}{r ccccccccccc}
\text{Chromosome} & A & G & C & C & T & A & G & A & T & T & C \\
\text{direct} & A & G & C & C & C & A & G & A & \phi & \phi & \phi \\
\text{direct} & A & G & C & C & T & A & G & N & T & - & \phi \\
\text{reverse} & \tilde{N} & \tilde{G} & \tilde{C} & \tilde{C} & \tilde{T} & \tilde{A} & \tilde{G} & \tilde{A} & \tilde{T} & \tilde{T} & \tilde{G} \\
\text{reverse} & \tilde{\phi} & \tilde{G} & \tilde{C} & \tilde{C} & \tilde{T} & \tilde{A} & \tilde{G} & \tilde{A} & \tilde{-} & \tilde{T} & \tilde{C} \\
\text{direct} & \phi & \phi & N & C & T & A & G & A & T & T & C
\end{array}$$

Fig. 1.1 An illustrative example of the problem. The bases with a tilde sign represent their complementary bases

1.2.2 Setup

Throughout the chapter, we represent random variables by capital letters and their values by small letters. First, reads are aligned into an assembly matrix. We introduce two alphabets: $\mathscr{A} = \{A, C, G, T, -\}$ and $\mathscr{B} = \{A, C, G, T, -, N, \phi\}$, where $-$ denotes an internal gap, N denotes any ambiguous determination of a base, and the null symbol ϕ is for non-aligned positions beyond the ends of a fragment. Each fragment is either in direct or in reverse complemented orientation. To deal with the issue of orientation, we introduce a complementary operation $\tilde{}$ on the alphabet $\mathscr{B}$ as follows: $\tilde{A} = T$, $\tilde{T} = A$, $\tilde{G} = C$, $\tilde{C} = G$, $\tilde{\phi} = \phi$, and $\tilde{-} = -$. An illustrative example of assembly matrices is shown in Fig. 1.1.

We denote the target sequence by $S = S_1 S_2 \ldots S_n$, where S_j takes any value from the alphabet $\mathscr{A}$. Random fragments generated from the template are aligned by an assembler. This results in an assembly matrix $\{X_{ij}\}_{m \times n}$. The elements of the fragment assembly matrix, denoted by x_{ij}, take values from the alphabet $\mathscr{B}$. Each row in $\{X_{ij}\}$ contains the ordered sequence of bases and possible gaps in a particular fragment. The column index $j = 1, \ldots, n$ runs from the leftmost base in the assembly to the rightmost. We represent the orientation of the i-th fragment in the assembly by

$$r_i = \begin{cases} 0 \text{ fragment } i \text{ is in direct orientation,} \\ 1 \text{ fragment } i \text{ is in reverse orientation.} \end{cases}$$

The observations $\{X_{ij}\}$ are subject to measurement error. We denote the true base of fragment i at position j by $Y_{ij} \in \mathscr{A}$. Therefore

$$Y_{ij} = \begin{cases} S_j \text{ if } r_i = 0\,, \\ \tilde{S}_j \text{ if } r_i = 1\,. \end{cases}$$

We denote the compositional probability by $\alpha_a = \Pr(S_j = a),\ a \in \mathscr{A}$.

1.2.3 Phred Quality Scores

After appropriate preprocessing, each sequencing chromatogram contains a series of peaks of four colors. The rationale of base-calling is that each peak represents one base, and the order of peaks from the four channels is consistent with the order of nucleotide bases on the underlying DNA fragment. In addition to base-calling, *Phred* also assigns each base-call a quality score q taking integer values from 0 to Q (Q is 64 for *Phred* scores) [5]. Quality scores are based on trace features such as peaking spacing, uncalled/called peak ratio and peak resolution. The model that defines quality scores was so trained on a large amount of sequencing traces that the scores could be interpreted as probabilities. Mathematically, the score is defined by

$$q_{ij} = -10\log_{10}\varepsilon_{ij}, \quad \text{where } \varepsilon_{ij} = \mathbf{Pr}(Y_{ij} \neq x_{ij}|X_{ij} = x_{ij}), \tag{1.1}$$

namely, ε_{ij} is the error probability of base-calling. Randomly select one position from an assembly and let Y and X be its true base and called base respectively. Let $\mathscr{E}$ be the event that the base-calling is incorrect, namely, $\mathscr{E} = \{X \neq Y\}$. Then the correct calling probability given base a is: $1 - \varepsilon = \mathbf{Pr}(Y = a|X = a)$, where $a \in \mathscr{A}$. Notice that

$$\mathbf{Pr}(X = a|Y = a) = \frac{\mathbf{Pr}(Y = a|X = a)\mathbf{Pr}(X = a)}{\mathbf{Pr}(Y = a)} = (1-\varepsilon)\frac{\mathbf{Pr}(X = a)}{\mathbf{Pr}(Y = a)}.$$

If the assumption of unbiased base-calling is valid, namely, $\mathbf{Pr}(X = a) = \mathbf{Pr}(Y = a)$, then we have $\mathbf{Pr}(X = a|Y = a) = \mathbf{Pr}(Y = a|X = a) = 1 - \varepsilon$. Consequently, we are able to interpret the *Phred* scores as probabilities by:

$$\mathbf{Pr}(X_{ij} = x_{ij}|Y_{ij} = x_{ij}) = \mathbf{Pr}(Y_{ij} = x_{ij}|X_{ij} = x_{ij}) == 1 - 10^{-q_{ij}/10}.$$

Even though *Phred* scores are valuable information for the construction of consensus, they are not the complete picture of measurement error. In general, for $a \neq b \in \mathscr{A}$, we have

$$\mathbf{Pr}(X = b|Y = a) = \mathbf{Pr}(X = b|Y = a, \mathscr{E})\mathbf{Pr}(\mathscr{E}|Y = a) = \mathbf{Pr}(X = b|Y = a, \mathscr{E})\cdot\varepsilon.$$

Denote sequencing error rates conditional on event $\mathscr{E}$ by $w(b|a) = \mathbf{Pr}(X = b|Y = a, \mathscr{E})$ for $a \neq b$, and we arrange them in the following table:

w	A	C	G	T	-
A		$w(C\|A)$	$w(G\|A)$	$w(T\|A)$	$w(-\|A)$
C	$w(A\|C)$		$w(G\|C)$	$w(T\|C)$	$w(-\|C)$
G	$w(A\|G)$	$w(C\|G)$		$w(T\|G)$	$w(-\|G)$
T	$w(A\|T)$	$w(C\|T)$	$w(G\|T)$		$w(-\|T)$
$-$	$w(A\|-)$	$w(C\|-)$	$w(G\|-)$	$w(T\|-)$	

,

where $\{w(b|a)\}$ satisfy

$$\sum_{b\in\mathscr{A},b\neq a} w(b|a) = 1 \quad \text{and} \quad w(b|a) \geq 0 \quad \text{for} \quad b \neq a.$$

The sequencing error rates relate to the conditional probabilities as follows.

$$\mathbf{Pr}(X = b|Y = a) = \begin{cases} \varepsilon\, w(b|a) & \text{if } a \neq b\,, \\ 1-\varepsilon & \text{if } a = b\,. \end{cases} \tag{1.2}$$

Since *Phred* scores provide only partial information about sequencing error rates, we need to estimate the rest. For the sake of simplicity, we skip the issue of fragment orientation when we describe the sequencing error models.

1.2.4 Conditional Sequencing Error Model

Our perspective is to incorporate *Phred* quality scores into the Churchill-Waterman model [3]. We first adopt the parameterization in (1.2) to model sequencing error, and refer to it as the conditional sequencing error model. The parameters θ in this model include the composition probability $\{\alpha_a\}$ and the conditional sequencing error rates $\{w(b|a)\}$. The likelihood of the assembly and underlying sequence is given by

$$\begin{aligned}&\left[\prod_{j=1}^{n}\prod_{i=1}^{m}\mathbf{Pr}(X_{ij} = x_{ij}|S_j;\theta)\right]\cdot\prod_{j=1}^{n}\mathbf{Pr}(S_j;\theta)\\ &\quad=\left[\prod_{j=1}^{n}\prod_{i=1}^{m}[(1-\varepsilon_{ij})^{\{\mathbf{1}(S_j=x_{ij})\}}\cdot(w(x_{ij}|s_j)\cdot\varepsilon_{ij})^{\{\mathbf{1}(S_j\neq x_{ij})\}}]\right]\cdot\prod_{j=1}^{n}\mathbf{Pr}(S_j;\theta)\,.\end{aligned}$$

Since $\{S_j\}$ are missing, we estimate the parameters by the E-M algorithm. The following form of log-likelihood is easy for imputing the sufficient statistics.

$$\begin{aligned}&\sum_{a\in\mathscr{A}}\left\{\sum_{b\in\mathscr{A}/\{a\}}\left[\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{1}(x_{ij} = b, S_j = a)\cdot\log[w(b|a)\,\epsilon_{ij}]\right]\right.\\ &\quad\left.+\left[\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{1}(x_{ij} = a, S_j = a)\cdot\log(1-\epsilon_{ij})\right]+\sum_{j=1}^{n}\mathbf{1}(S_j = a)\log\alpha_a\right\}.\end{aligned} \tag{1.3}$$

1.2.5 Mixture of Logistic Model

From a regression perspective, we take *Phred* scores as a covariate. Denote

$$\mu(b|a, q_{ij}) = \mathbf{Pr}(X_{ij} = b|S_j = a; q_{ij}), \quad a, b \in \mathscr{A}\,. \tag{1.4}$$

We assume that base-calling error rates follow a logistic form:

$$\log\left(\frac{\mu(b|a,q)}{\mu(a|a,q)}\right) = \beta_{a,b,0} + \sum_{l=1}^{L} \beta_{a,b,l}\, h_l(q)\,, \quad b \in \mathscr{A}/\{a\}\,,$$

where each covariate $h_l(q)$ is a function of quality score q and takes the form

$$h_l(q) = (q - o_l)_+ = \begin{cases} 0\,, & \text{if } q \le o_l, \\ q - o_l\,, & \text{otherwise.} \end{cases}$$

Notice that each function has a knot o_l, where $0 \le o_1 < o_2, \ldots < o_L < Q$. Thus each regressor is a piecewise linear function of quality score. It allows us to approximate any potential nonlinear effect. Equivalently, base-calling rates can be represented as:

$$\begin{cases} \mu(b|a,q) = \frac{\exp\{\beta_{a,b,0} + \sum_{l=1}^{L} \beta_{a,b,l}\, h_l(q)\}}{1 + \sum_{c \in \mathscr{A}/\{a\}} \exp\{\beta_{a,c,0} + \sum_{l=1}^{L} \beta_{a,c,l}\, h_l(q)\}}, & b \in \mathscr{A}/\{a\}, \\ \mu(a|a,q) = \frac{1}{1 + \sum_{c \in \mathscr{A}/\{A\}} \exp\{\beta_{a,c,0} + \sum_{l=1}^{L} \beta_{a,c,l}\, h_l(q)\}}. & \end{cases}$$

Similarly to (1.3), this parameterization leads to the following form of log-likelihood function for the assembly and the underlying sequence, up to a term only relating to parameters.

$$\begin{aligned} \sum_{a \in \mathscr{A}} \Bigg\{ & \sum_{b \in \mathscr{A}/\{a\}} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{1}(x_{ij} = b, S_j = a; q_{ij}) \\ & \cdot \log \frac{e^{\{\beta_{a,b,0} + \sum_{l=1}^{L} \beta_{a,b,l}\, h_l(q_{ij})\}}}{1 + \sum_{c \in \mathscr{A}/\{a\}} e^{\{\beta_{a,c,0} + \sum_{l=1}^{L} \beta_{a,c,l}\, h_l(q_{ij})\}}} \\ & + \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{1}(x_{ij} = a, S_j = a; q_{ij}) \\ & \cdot \log \frac{1}{1 + \sum_{c \in \mathscr{A}/\{a\}} e^{\{\beta_{a,c,0} + \sum_{l=1}^{L} \beta_{a,c,l}\, h_l(q_{ij})\}}} \\ & + \sum_{j=1}^{n} \mathbf{1}(S_j = a) \log \alpha_a \Bigg\}, \end{aligned}$$

where θ represents all the unknown parameters.

1.2.6 Parameter Estimation and E-M Training Algorithm

In both the conditional sequencing error model and the logistic model, the underlying sequence $\{s_j\}$ is unknown. Its reconstruction relies on the estimates of parameters in the models. On the other hand, algorithms of estimating parameters are well established when $\{s_j\}$ are known. Thus we use the E-M algorithm to train the model iteratively. In the E-step, we impute the sufficient statistic from observations at the current parameter value; In the M-step, we update the maximum likelihood estimate using the current imputed values of missing data. In the case of the conditional sequencing error model, the parameters are estimated by counting frequencies. In the case of the logistic model, the likelihood can be decomposed into five independent logistic regression models; see [18]. Consequently, we run re-weighted least squares to estimate the parameters [23].

1.2.7 Consensus and Quality Values

According to the logistic model, the distribution of nucleotides at each position is given by:

$$\begin{aligned}&\mathbf{Pr}(S_j = a|\{X_{ij} = x_{ij}\};\{q_{ij}\})\\ &\quad = \frac{\alpha_a \prod_{i=1}^{m}\big[(1-r_i)\cdot\mu(x_{ij}|a,q_{ij}) + r_i\cdot\mu(\tilde{x}_{ij}|\tilde{a},q_{ij})\big]}{\sum_{b\in\mathscr{A}}\alpha_b \prod_{i=1}^{m}\big[(1-r_i)\cdot\mu(x_{ij}|b,q_{ij}) + r_i\cdot\mu(\tilde{x}_{ij}|\tilde{b},q_{ij})\big]}.\end{aligned}$$

As shown in the formula, the issue of orientation can generally be dealt with by the orientation indicators $\{r_i\}$ and the complementary operator $\tilde{}$. In our convention, we observe $\tilde{x}_{ij}$ directly when a fragment is in reverse orientation. After we plug in the estimated value of θ, we define the consensus at one position and its quality score by maximizing the above probability.

1.2.8 Parsimonious Representation and Model Selection

Although we can include piecewise linear functions at all possible knots in the logistic regression model (1.4), we seek a parsimonious model for several purposes. First, we would like to avoid potential over-fitting, especially when the size of assembly is not large. Second, a parsimonious model may give us insights into quality scores.

The selection of knots is nothing but a model selection problem. To compare different models, we need an evaluation criterion. Based on quality scores, each fitted model defines a set of error rates, which in turn can be used to construct a consensus. If the truth is known, we can calculate single base discrepancy (SBD) for a model, see [7]. SBD is thus one criterion for model comparison.

A practical solution to model selection ought to be self-evident from data. One such criterion is BIC (Bayesian information criterion) [22]. It is defined as

$$BIC = -\text{log-likelihood of assembly} + \frac{1}{2}(\#\ \text{parameter}) \log(\#\ \text{observation})\,.$$

Namely, BIC penalizes log-likelihood by model complexity in terms of number of parameter. For a logistic model with L knots, we have $20(L+1)$ parameters. We calculate the BIC score for each model, and choose the one that minimizes the quantity. The idea is to trade off goodness of fit and model complexity. Computationally, it is intensive to evaluate every model. We adopt the backward deletion strategy used in regression analysis to search for the optimal model [23]. This is motivated by the fact that backward deletion strategy coupled with BIC leads to consistent model estimates in the case of linear regression [2].

1.2.9 Bias of Quality Scores

If we do not specify data source otherwise, the results reported hereafter are based on the *Campylobacter jejuni* sequencing data explained earlier. In the conditional sequencing error model, quality scores are interpreted as error probabilities of base-calling. The model that defines the *Phred* scores is determined from a training data set, see [5,6]. When the model is applied to sequencing traces obtained under different working conditions, scores may deviate from probabilities to some extent. We examine this issue on sequencing reads from one BAC. After alignment, we count incorrect base-calls for each value of quality scores – *Phred* scores take integer values from 0 to 64. The observed score for predicted quality score q is calculated from the assembly by:

$$q_{obs}(q) = -10 \cdot \log_{10}\left(\frac{Err_q}{Err_q + Corr_q}\right),$$

where Err_q, $Corr_q$ are respectively the number of incorrect and correct base-calls at quality score q. In Fig. 1.2 we plot the observed scores against predicted *Phred* scores. When scores are above 55, essentially no error are observed. When scores are below 20, the prediction is fairly consistent. When scores are between 20 and 55, *Phred* scores overestimate probabilities. Thus calibration is desired for the purpose of improving accuracy of base-calling. Next we apply the logistic model to the data. Let

$$\epsilon'_{ij} = \mathbf{Pr}(S_j \neq x_{ij} | X_{ij} = x_{ij}, q_{ij}; \theta),$$

where θ represents all the parameters. Under the assumption of unbiased base-calling, we have:

$$\epsilon'_{ij} = 1 - \mathbf{Pr}(X_{ij} = x_{ij} | S_j = x_{ij}, q_{ij}; \theta) = 1 - \mu(x_{ij} | x_{ij}, q_{ij}).$$

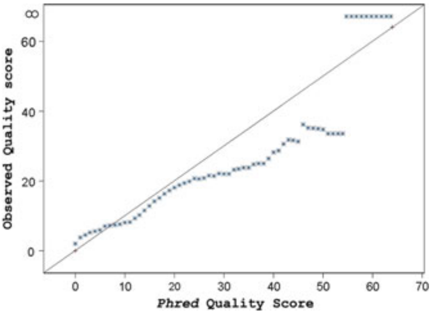


Fig. 1.2 Observed sequencing error rates vs. predicted error rates by *Phred* Quality Score

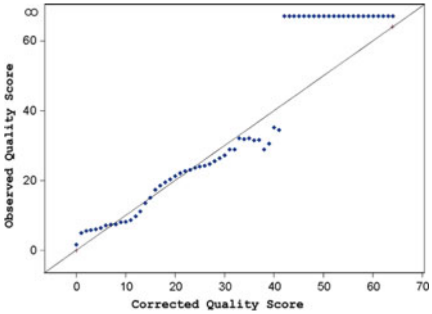


Fig. 1.3 Observed sequencing error rates vs. predicted error rates by a mixture of logistic model

Then we can assign a new quality score to each base-call x_{ij}:

$$q'_{ij} = -10 \cdot \log_{10} \epsilon'_{ij} .$$

The bias of this adjusted quality score can be examined by:

$$q_{obs}(q') = -10 \cdot \log_{10} \left(\frac{Err_{q'}}{Err_{q'} + Corr_{q'}} \right),$$

where $Err_{q'}$ and $Corr_{q'}$ are respectively the number of incorrect and correct base-calls at adjusted quality score q'. We plot the observed against the corrected quality score in Fig. 1.3. Compared with Fig. 1.2, we see that the corrected quality score is more consistent with the observed quality score. After adjustment, no error occurs above score value 42.

The CEQ software coming along with Beckman sequencers offer quality values similar to *Phred* scores; see [24]. However, their scores are trained from a smaller data set compared to *Phred*. Like Fig. 1.2 we plot the observed scores against predicted *CEQ* scores in Fig. 1.4. The overestimate pattern is seen across almost the entire region. Then we apply the adjustment procedure to correct for the obvious bias. The training data set is about 500 kb. In Fig. 1.5 we plot the observed against the corrected quality scores.

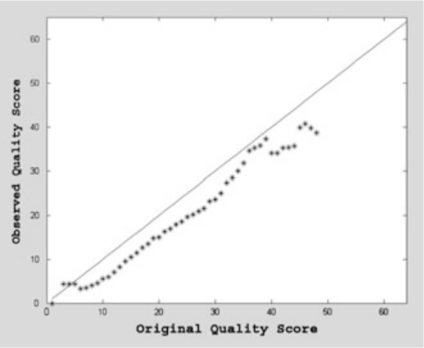


Fig. 1.4 Observed sequencing error rates vs. predicted error rates by CEQ Quality Score

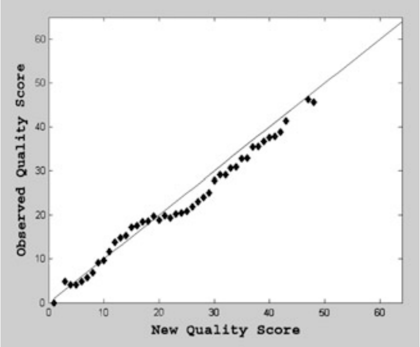


Fig. 1.5 Observed sequencing error rates vs. predicted error rates by a mixture of logistic model

1.2.10 Score-Dependent Error Patterns

In the conditional sequencing error model, we assume that the error patterns, or the conditional error rates, are constant regardless of quality scores. To check the assumption, we compare the frequencies of each type of sequencing errors at each quality value ranging from 0 to 64. That is, given a assembly, we calculate the empirical conditional error rates as follows,

$$w_{obs}(b|a;\, q) = \frac{\sum_{i,j} \mathbf{1}(x_{ij} = b, s_j = a, q_{ij} = q)}{\sum_{c \in \mathscr{A}/\{a\}} \sum_{i,j} \mathbf{1}(x_{ij} = c, s_j = a, q_{ij} = q)}, \quad a, b \in \mathscr{A}.$$

When the true base is A, we plot these conditional error rates against quality scores in Fig. 1.6. It indicates that error patterns do depend on quality scores. After we fit a logistic model to the assembly, the conditional error probabilities as a function of quality scores are shown in Fig. 1.7. When quality scores are above 55, no sequencing error is observed. Thus conditional error patterns make sense only for scores below 55. Many sequencing projects use the Q_{20} rule as a rough measure of the effective length of a DNA read; see [19] for more discussions. Scores below 20 indicate low quality regions. As we can see, error patterns change significantly at around 20–24. Since we do not have many bases with high scores, the inference in

Fig. 1.6 Observed score-wise conditional error rates. The true base is A

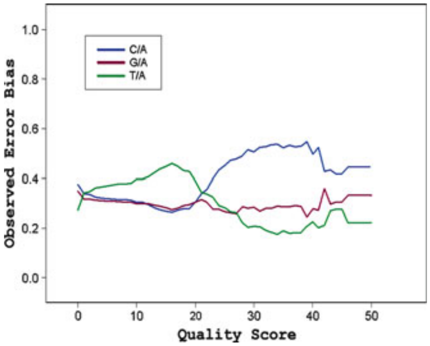
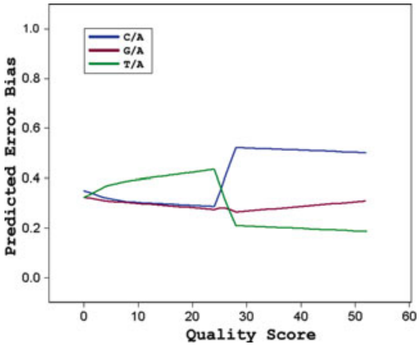


Fig. 1.7 Conditional error rates predicted from a logistic model. The true base is A



the high quality range is less reliable than that in the low quality range. When quality score are below 20, C and G are similar to each other; when the scores are beyond 24, a totally different error pattern is observed. This by-product of the parsimonious model offers another perspective of the Q_{20} rule.

1.2.11 Comparison of Different Methods

We have introduced a conditional sequencing model and a logistic model. In the literature two quite different methods exist to estimate sequencing error rates. On the one hand, the method proposed by Churchill and Waterman [3] relies only on an assembly but not quality scores, and we refer to it as the simple probability model. On the other hand, we can use *Phred* scores and simply assign equal error chances among bases. Hereafter we refer to it as the simple quality score method. In Table 1.1, we compare the performance of these methods on the *C. jejuni* data set. The majority rule defines the consensus by choosing the most frequent nucleotide at each position. Compared with the majority rule, the simple probability model reduces errors by one quarter, not resorting any other information other than the assembly itself. The simple quality score method cuts errors to more than half. The gain is from the training data set that defines *Phred* scores. The conditional

Table 1.1 Comparison of different methods. The majority rule is a straightforward counting strategy; The simple probability model is the method proposed by Churchill and Waterman [3]; The simple quality score method uses *Phred* scores and assigns equal error chances among bases; Conditional Sequencing model uses *Phred* scores and estimate error pattern from data by an E-M algorithm; Logistic Model predicts sequencing errors by *Phred* scores

Method	Single base discrepancy	Log-likelihood of assembly
Majority rule	810	
Simple probability	591	−339,704
Simple quality score	367	−292,781
Conditional sequencing error	358	−281,411
Logistic (5 knots)	346	−272,341

sequencing error model reduces errors further. And the best result, 346 SBDs, is achieved by the logistic model with five knots. BIC selects a logistic model with three knots that has 348 SBDs. The likelihood scores of these models are also shown in Table 1.1. The likelihood of a model measures its goodness of fit to the data. For the same data set, we slightly perturb the *Phred* scores associated with the called bases, and errors resulted from the simple quality score method increase substantially from 367 to 517 while the performance of logistic method remains almost the same.

1.3 Reconstruction of Diploid Consensus Sequences and its Accuracy Assessment

One of the main goals in genome sequencing projects is to determine a haploid consensus sequence even when clone libraries are constructed from homologous chromosomes. However, it has been noticed that haplotypes can be inferred from genome assemblies by investigating phase conservation in sequenced reads. Next we describe a method that directly seeks to infer haplotypes, a diploid consensus sequence, from the genome assembly of an organism,

1.3.1 The Probabilistic Model

Consider n potential polymorphic sites and m fragments. The notation in this section is slightly different from that in last section. We denote two target chromosomes by $\mathbf{S} = \{S_{k1}S_{k2}\ldots S_{kn},\ k = 1,2\}$. Each letter takes values from the alphabet $\mathscr{A} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}, -, \mathtt{M}\}$, where $-$ denotes an internal gap, and M denotes any sequence of two or more nucleotide bases. For simplicity, we assume that the genotypes are independently and identically sampled from the composition probabilities

$$\mu(a,b) = \Pr(S_{1,j} = a,\ S_{2,j} = b) = \Pr(S_{1,j} = b,\ S_{2,j} = a),\ a,\ b \in \mathscr{A}.$$

The origins of the fragments are denoted by $\mathbf{F} = \{F_i,\ i = 1,\ldots,m\}$, and they appear according to Bernoulli trials:

$$F_i = \begin{cases} 1 \text{ with prob } \lambda_1 = \lambda\,, \\ 2 \text{ with prob } \lambda_2 = 1 - \lambda\,. \end{cases}$$

We denote the true bases of the assembly matrix by $\mathbf{Y} = \{Y_{ij},\ i = 1,\ldots,m,\ j = 1,\ldots,n\}$, and they relate to the target haplotype by: $Y_{ij} = S_{F_i,j}$, $i = 1,2$. The observations $\mathbf{X} = \{X_{ij},\ i = 1,\ldots,m,\ j = 1,\ldots,n\}$ are the measurement of $\mathbf{Y}$ via the following random error model:

$$\eta_{ij}(b|a) = \Pr(X_{ij} = b|Y_{ij} = a),\ a \in \mathscr{A},\ b \in \mathscr{B},$$

where $\mathscr{B} = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}, -, \mathtt{M}, \phi\}$, the null symbol ϕ denotes any ambiguous determination of a base or positions beyond the ends of a fragment. The errors can be categorized as single-nucleotide replacement, single-nucleotide insertion, deletion, and errors involving multiple nucleotides. Sometimes we drop the subscript of η when no confusion is incurred. We have assumed that measurement errors occur independently with identical distribution across the assembly because the notation is complicated without the assumption. The random fragments are generated either in the direct or reversed orientation. To deal with the issue, we introduce the complementary letters as follows: $\tilde{\mathtt{A}} = \mathtt{T}$, $\tilde{\mathtt{T}} = \mathtt{A}$, $\tilde{\mathtt{G}} = \mathtt{C}$, $\tilde{\mathtt{C}} = \mathtt{G}$, $\tilde{\mathtt{M}} = \mathtt{M}$, $\tilde{\phi} = \phi$, and $\tilde{-} = -$. In Fig. 1.8, we illustrate the data structure by a hypothetical example. For the sake of notational simplicity, we skip the issue of orientation and nonpolymorphic sites. The two target chromosomes are shown at the top and bottom respectively. Six fragments are aligned in the middle.

In reality only the assembly matrix $\{x_{ij}\}$ is observed while the information of $\mathbf{S}$ and $\mathbf{F}$ is missing. Thus we need to estimate $\mathbf{S}$ and $\mathbf{F}$ based on the observations. Technically, the estimation of $\mathbf{S}$ can be based on its conditional distribution given data: $\Pr(\mathbf{S}|\mathbf{X})$. According to the Bayes' rule, we have

$$\Pr(\mathbf{S}|\mathbf{X}) = \frac{\Pr(\mathbf{X},\mathbf{S})}{\Pr(\mathbf{X})}\,, \tag{1.5}$$

Chromosome 1	A	G	C	C	M	A	G	A	T	T	C
Origin 1	A	G	A	C	M	A	G	A	ϕ	ϕ	ϕ
2	C	C	T	A	$-$	G	C	T	A	ϕ	ϕ
1	ϕ	G	C	C	M	A	G	A	T	T	ϕ
2	C	C	T	A	$-$	T	C	T	A	G	T
2	ϕ	C	T	A	$-$	G	C	T	$-$	G	T
1	ϕ	ϕ	ϕ	C	M	A	G	A	C	T	C
Chromosome 2	C	C	T	A	$-$	G	C	T	A	G	T

Fig. 1.8 An illustrative example of the problem. The two target chromosomes are shown at the top and bottom respectively. The fifth polymorphic site "M", in this case, represents "CCC". Six fragments are aligned in the middle. In reality, the targets and origins of fragments are not observed

where $\Pr(\mathbf{X}) = \sum_{\mathbf{S}} \Pr(\mathbf{X}, \mathbf{S})$. The formula to compute $\Pr(\mathbf{X}, \mathbf{S})$ is given by

$$\Pr(\mathbf{X}, \mathbf{S}) = \Pr(\mathbf{S}) \Pr(\mathbf{X}|\mathbf{S}) = [\prod_{i=1}^{m} \Pr(S_{1j},\ S_{2j})][\prod_{i=1}^{m} \sum_{k=1}^{2} \lambda_k \prod_{j=1}^{n} \Pr(X_{ij}|S_{kj})]\,. \tag{1.6}$$

We define the most probable haplotypes by: $\max_{\mathbf{S}} \Pr(\mathbf{S}|\mathbf{X})$. It is possible that we cannot determine all the phase information because the coverage and origins of fragments are not strictly uniform across the entire clone. Thus we look for relatively shorter haplotype segments that exceed some level of confidence. The calculation of the confidence score for a given haplotype configuration is also based on (1.5) and (1.6). However, the marginal probability $\Pr(\mathbf{X})$ is the sum of joint probabilities over all haplotypes. The complexity of a straightforward algorithm is $O(5^{2n})$. Next we develop an algorithm of linear complexity with respect to the number of polymorphic sites.

1.3.2 A Markov Structure

We start off with one locus and then move along the chromosome recursively. Suppose we have dealt with $k - 1$ loci and are considering the k-th locus. We notice that only fragments that cover the position are relevant. Denote the index set of those fragments covering the k-th locus by $\Omega(k)$. We note that only these fragments are relevant for the calculation. Let $\Theta(k) = \bigcup_{j=1}^{k} \Omega(j)$. We decompose $\Omega(k)$ into four subsets: $\Lambda_1(k)$ includes those fragments covering the k-th locus but neither the $(k-1)$-th nor the $(k+1)$-th; $\Lambda_2(k)$ includes those fragments covering both the $(k-1)$-th and k-th locus but not the $(k+1)$-th; $\Lambda_3(k)$ includes those fragments covering both the k-th and $(k+1)$-th locus but not the $(k-1)$-th; $\Lambda_4(k)$ includes those fragments covering the $(k-1)$-th, k-th and $(k+1)$-th locus. We write $\Gamma(k) = \Lambda_3(k) \bigcup \Lambda_4(k)$. An illustration of the definition is shown in Fig. 1.9. It is easy to check that $\Gamma(k) = \Lambda_2(k+1) \bigcup \Lambda_4(k+1)$, and $\Theta(k+1) = \Theta(k) \bigcup \Lambda_1(k+1) \bigcup \Lambda_3(k+1)$. Figure 1.10 shows how the index sets evolve as the calculation moves along a clone.

If we compute likelihood iteratively along the chromosome, then we need the dependence structure of $\Omega(k+1)$ on $\Theta(k)$. According to the definition of $\Gamma(k)$, we have the following.

Proposition 1 $\{S_{k,1} = a_1, S_{k,2} = a_2\ , F_i = f_i,\ i \in \Gamma(k)\}$ *is a Markov chain.*

It is interesting to see that the dimension of this state vector varies across loci. We define

$$\begin{aligned}
&\alpha_k(a_1, a_2; f_i,\ i \in \Gamma(k)) \\
&\quad = \Pr(\mathbf{X}_{ij} = x_{ij},\ j = 1, \ldots, k,\ i = 1, \ldots, m; S_{k,1} = a_1, S_{k,2} = a_2; \\
&\qquad F_i = f_i,\ i \in \Gamma(k))\,.
\end{aligned}$$

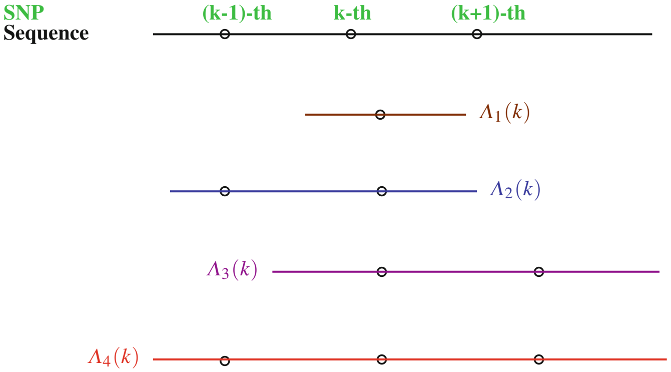


Fig. 1.9 Definition of four index sets. $\varGamma(k) = \varLambda_3(k) \bigcup \varLambda_4(k) = \varLambda_2(k+1) \bigcup \varLambda_4(k+1)$

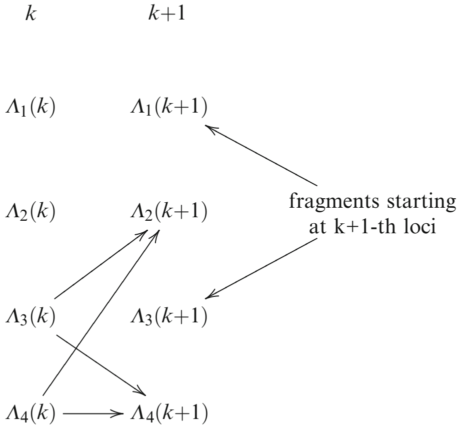


Fig. 1.10 An illustration of the recursive structure of the index sets. Formula (1.7) uses this structure

The hidden Markov model is widely used in many areas such as speech recognition and computational biology because fast algorithms exist for modeling and decoding. For example, the likelihood of data for a hidden Markov model can be evaluated by the forward-backward algorithm, whose complexity is linear with respect to time. The above definition is motivated by the forward-backward algorithm in the hidden Markov model. Based on the above Markov structure, we can recursively compute $\alpha_k(a_1, a_2; f_i,\ i \in \varGamma(k))$ as follows.

Theorem 1

$$\alpha_{k+1}(a_1, a_2; f_i,\ i \in \varGamma(k+1)) = \mu(a_1, a_2) \left[\prod_{h=1}^{2} \lambda_h^{\sum_{j \in \varLambda_3(k+1)} \mathbf{1}(F_j = h)} \right]$$

$$\left[\prod_{j\in\Lambda_1(k+1)} [\lambda_1\, \eta(x_{j,k+1}|a_1) + \lambda_2\, \eta(x_{j,k+1}|a_2)]\right]$$
$$\left[\prod_{j\in\Lambda_3(k+1)} \eta(x_{j,k+1}|a_{f_j})\right]\left[\prod_{j\in\Lambda_4(k+1)} \eta(x_{j,k+1}|a_{f_j})\right]$$
$$\left[\sum_{f_j=1,2,\ j\in\Lambda_2(k+1)} \left[\prod_{j\in\Lambda_2(k+1)} \eta(x_{j,k+1}|a_{f_j})\right] \sum_{b_1,b_2} \alpha_k(b_1,b_2; f_j,\ j\in\Gamma(k))\right] \tag{1.7}$$

Please notice that if $\Lambda_2(k+1)$ is empty, then we skip the corresponding summation in the formula. If at position d, the set $\Gamma(d)$ is empty, then we have

$$\Pr(\mathbf{X}_{ij} = x_{ij},\ i = 1,\ldots,m, j = 1,\ldots,d) = \sum_{a_1,a_2} \alpha_d(a_1,a_2)\,. \tag{1.8}$$

The proof can be found in [8]. Despite the appearance of the formula, we can prove the following result.

Proposition 2 *The complexity of the algorithm defined by (1.7) and (1.8) is linear with respect the number of polymorphic sites. The expected complexity is proportional to e^{κ}, where κ is the average coverage.*

This is true because in each step of the recursion we deal with one more locus by (1.7) and keep the state variables $\{\alpha_k(a_1,a_2; f_i,\ i\in\Gamma(k))\}$ in memory. The memory size is thus the state dimension, which is not constant along a chromosome. Denote the coverage variable by K. Approximately, it follows a Poisson distribution with the parameter κ. On average, the memory size is proportional to $E[2^K] = E[e^{\log 2\, K}] = e^{\kappa}$ according to the moment generating function of Poisson distribution.

The peak memory usage is an important issue in practice. Of course, we can keep the active coverage in some manageable range by randomly skipping some fragments. Next we evaluate the worst case by the Poisson distribution

$$\Pr(K \geq m) = \sum_{j=m}^{\infty} \frac{\kappa^j e^{-\kappa}}{j!}\,.$$

Let $W_1, W_2, \cdots, W_m$ be independent and exponentially-distributed random variables with parameter κ. According to the structure of the Poisson process, [21], the above quantity equals

$$\Pr(K \geq m) = \Pr(W_1 + W_2 + \cdots + W_m \leq 1) = \int_o^1 \frac{\kappa^m t^{m-1} e^{-\kappa t}}{(m-1)!}\, dt\,,$$

namely, an incomplete Gamma integral. In the case of *Ciona intestinalis* shotgun sequencing, κ is seven, and $\Pr(K \geq 20) = 0.000044$, $\Pr(K \geq 23) \leq 10^{-6}$. Thus it is very unlikely that the memory requirement exceeds 2^{20}. Our simulations justify this analysis.

1.3.3 Sequencing Error Rates and Quality Scores

The values of $\eta_{ij}(b|a)$ are crucial in our reconstruction procedure. If we assume that base-calling is independent and identically distributed across the assembly and $\eta_{ij}(b|a) = \eta(b|a)$, then we can apply an E-M procedure similar to that in [3] to estimate them.

Another approach makes use of the quality scores provided by some base-calling algorithms [5]. We can connect these scores to our model if a valid probabilistic interpretation is available. First we consider the cases of SNPs. Write

$$\varepsilon_{ij} = \Pr(\text{Base-call} \neq a|\text{True base} = a).$$

The quality scores are usually given in the following transformed form:

$$q_{ij} = -10\log_{10}\varepsilon_{ij}\,.$$

Approximately we have

$$\eta_{ij}(b|a) = \begin{cases} 1-\varepsilon_{ij} & a = b\,, \\ \varepsilon_{ij}\omega(b|a) & a \neq b. \end{cases}$$

The error-bias parameters $\{\omega(b|a),\ a \neq b,\ a \in \mathscr{A},\ \ b \in \mathscr{B}\}$ can either be set to some constants or can be estimated from data. In the case of complex indels, we align an observed sequence with a template, and calculate $\eta_{ij}(b|a)$ by multiplying scores from each position. In the haploid case, the last section provided a method to calibrate quality scores and estimate conditional error probability using mixture of logistic regressions.

1.3.4 Reconstruction of Diploid Genome

The conditional probability $\Pr(\mathbf{S}|\mathbf{X})$ plays a key role in our reconstruction, and we term it as confidence score. Due to the computational complexity, we proposed a pairwise strategy to find the most probable haplotype configuration [14]. That is, we start by considering each adjacent pair. To determine the haplotype for two loci, we check the odds ratio of the two most probable states and the pairwise confidence score. In addition to forward linking of adjacent loci, we also check confidence scores and adjust solutions in a backward fashion.

Algorithm 1 Reconstruction of haplotype segments

1. *For each locus, we report the most probable genotype according to the single confidence score.*
2. *For each adjacent pair, we report the most probable haplotypes according to the odds ratio and pairwise confidence score.*
3. *Link the haplotype phases obtained in Step 2 and construct haplotype segments. If inconsistent adjacent pairs occur, then we consider these sites jointly.*
4. *Evaluate the overall confidence score for each haplotype segment.*
5. *If the confidence score is below a threshold, we check the pair of loci with the smallest pairwise confidence score obtained in step 2. Then we compute the overall confidence score by flipping the phase between these two loci. If the score exceeds the threshold, we stop; otherwise we break the segment into two. In this case, we repeat the step 4 and 5 respectively to these two segments.*

In the case that the nominal frequency value is known, we can still use its estimated value in the calculation of confidence scores to achieve adaptive reconstruction and consequently to improve accuracy of reconstruction.

1.3.5 Mate-Pair Information and Second-Stage Bridging

The above model applies to the case of two-end sequencing if we assign the letter ϕ to those un-called bases in the middle of each fragment. However, when we evaluate the overall confidence score by (1.7) and (1.8), a clone remains active even at those polymorphic sites between the two ends. This may increase the coverage by several folds. We notice that the mate-pair information does not provide much extra help in regions (scaffolds) of high coverage. Thus we skip the mate-pair information in the first round of Algorithm 1. If two contigs are connected by at least one clone through mate-pair fragments after the first round, then we apply the algorithm to the two contigs and "bridging" clones, trying to determine the phase. We present some details in what follows. Suppose one contig consists of two haplotype segments $S_1^{(L)}$ and $S_2^{(L)}$, another contig consists of $S_1^{(R)}$ and $S_2^{(R)}$. Two possible phase configurations between the two non-overlapping contigs are shown in Fig. 1.11 and we denote them by C_1 and C_2 respectively. Suppose some clones overlap with one contig at one end and overlap with another at the other end. Denote these clones by $\mathbf{Z} = \{Z_i, i \in I\}$. For each clone Z_i, denote by $J_{i,L}$ the index set of those polymorphic sites that overlap with $S_1^{(L)}$ and $S_2^{(L)}$. Similarly we define $J_{i,R}$ for the polymorphic sites on Z_i at the other end. Thus $Z_i = \{Z_{ij}, j \in J_{i,L}\} \cup \{Z_{ij}, j \in J_{i,R}\}$. Then

$$\Pr(\mathbf{Z}|C_1) = \prod_{i \in I} \Pr(Z_i|C_1), \quad \Pr(\mathbf{Z}|C_2) = \prod_{i \in I} \Pr(Z_i|C_2),$$

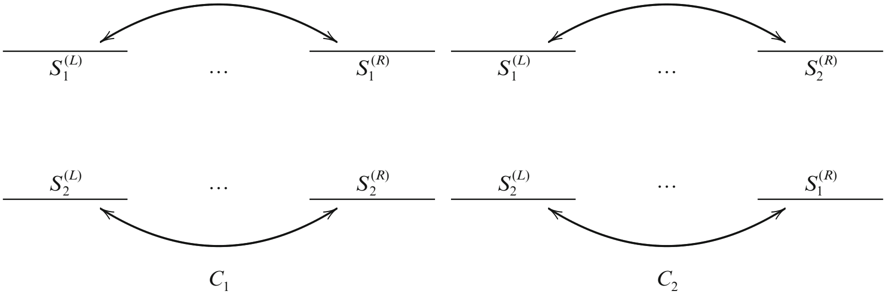


Fig. 1.11 Bridge two contigs by mate-pair fragments. Two possible configurations are shown

where

$$\begin{aligned}\Pr(Z_i|C_1) &= \frac{1}{2}\prod_{j\in J_{i,L}}\Pr(Z_{ij}|S_{1j}^{(L)})\prod_{j\in J_{i,R}}\Pr(Z_{ij}|S_{1j}^{(R)})\\ &\quad+\frac{1}{2}\prod_{j\in J_{i,L}}\Pr(Z_{ij}|S_{2j}^{(L)})\prod_{j\in J_{i,R}}\Pr(Z_{ij}|S_{2j}^{(R)})\\ \Pr(Z_i|C_2) &= \frac{1}{2}\prod_{j\in J_{i,L}}\Pr(Z_i j|S_{1j}^{(L)})\prod_{j\in J_{i,R}}\Pr(Z_{ij}|S_{2j}^{(R)})\\ &\quad+\frac{1}{2}\prod_{j\in J_{i,L}}\Pr(Z_{ij}|S_{2j}^{(L)})\prod_{j\in J_{i,R}}\Pr(Z_{ij}|S_{1j}^{(R)})\,.\end{aligned}$$

Based on the calculation, we make the following decision according to a threshold larger than one:

$$\begin{cases}\text{accept } C_1 & \text{if } \frac{\Pr(\mathbf{Z}|C_1)}{\Pr(\mathbf{Z}|C_2)} > \text{threshold}\,,\\ \text{accept } C_2 & \text{if } \frac{\Pr(\mathbf{Z}|C_2)}{\Pr(\mathbf{Z}|C_1)} > \text{threshold}\,,\\ \text{no decision if otherwise}\,. & \end{cases}$$

We iterate this bridging step to extend haplotype segments.

To evaluate the confidence score of any extended contig, we regard a "bridging" clone as one single fragment by including its mate-pair information. We emphasize that the two mate-pair fragments of a clone are not considered jointly if they fall in the same contig because the extra phase information is negligible in this case. Only the two mate-pair fragments from a "bridging" clone are treated as linked in formula (1.7). Thus coverage is not an issue any more.

1.3.6 Inference of Haplotype Frequency

Next our focus turns to the issue of haplotype frequency. For any fixed value of λ, the recursive algorithm (1.7) and (1.8) allows us to efficiently compute the probability of the fragment assembly, or the likelihood as called in the inference theory. Denote the log-likelihood of the observed assembly by

$$L(\lambda) = \log \Pr(\mathbf{X}; \lambda) ,$$

where **X** is the assembly matrix. By maximizing the log-likelihood with respect to the haplotype frequency parameter in the range from 0 to 1/2, we can obtain its estimate. In general, the maximum likelihood estimate is asymptotically efficient under regularity conditions. In other word, it is one of the most accurate estimates in the large sample scenario. Alternatively, if we have several hypothetical values for the haplotype frequency known from a genomic or genetic context, say, $\lambda = 0, 1/4, 1/3, 1/2$, then we can select the value that achieves the largest likelihood. In real genome assembly, we can use the estimate of haplotype frequency to monitor the existence of misalignment of "bad" fragments.

1.3.7 An Example

Ciona intestinalis is an important organism to study the origins of chordates and vertebrates. A draft of its protein-coding portion has been reported [4], Its high polymorphism rate, about 1.2% as reported, makes it an ideal case for reconstructing haplotypes from shotgun sequencing. To evaluate the proposed methodology, we simulated contigs according to the parameters obtained from *Ciona* sequencing.

The simulation was based on the stochastic model proposed in [11]. Denote the clone length by H. According to the random model, the number of polymorphic sites in the clone, denoted by $N(H)$, is a Poisson random variable with the parameter λH, where $1/\lambda$ measure the average inter-arrival distance between adjacent polymorphic sites. Conditional on the total number, the positions of polymorphic sites are uniformly distributed along the interval [0, H]. We generated random fragments ($1.8 \sim 120$ K bp) according to their proportions in the *Ciona intestinalis* [4] and simulated two-end sequencing of the fragments. The average sequencing read was 650 bp, and coverage was seven. To match the polymorphism rate of *Ciona intestinalis*, the expected inter-arrival time between potential adjacent loci was set to be 66 bp. The sequencing error rates were about 4%.

We reconstructed haplotype segments by applying Algorithm 1. In step 2, we scanned each adjacent pair of loci for significant haplotypes. The threshold was set as follows: the pairwise confidence score is larger than 0.5 and the odds ratio of the top two most probable cases is larger than 1.1. We reported outcomes under two haplotype frequencies, 0.5 and 0.25. The results are shown in Table 1.2. In the case of $r = 0.5$, the true positive rate was 97.05%. The percentage of correctly

Table 1.2 The reconstruction result from a simulation based on *Ciona intestinalis*. The polymorphism rate is 1.2%. The total size of scaffolds is about 60 M bp. To determine the significance of pairwise comparison, we set the thresholds for pairwise confidence to be 0.5 and odds ratio of the two most probable cases to be 1.1. The number of polymorphisms in the final report includes singletons, namely, those single sites that cannot be connected to others. In this case, we report their genotypes. The true positive rates are for those reported sites, either genotypes or haplotypes. The last two accounts are the percentage of correctly detected pairs among all the polymorphic sites generated and average lengths of haplotype segments

Haplotype frequency	$\lambda = 1/2$	$\lambda = 1/2$	$\lambda = 1/4$
Mate-pair information	w/o mate-pair	w/i mate-pair	w/i mate-pair
Total # polymorphism	674,246	674,246	671,359
# polymorphism reported (including singleton)	618,034	618,034	554,442
True positive rate (all reported)	97.5%	97.1%	96.1%
Percentage of correctly detected sites	89.4%	89.0%	79.4%
Average segment length	45.4	70.4	31.1

Table 1.3 The reconstruction result from a simulation of a polymorphism rate 0.3%, *cf.* Table 1.2

Haplotype frequency	$\lambda = 1/2$	$\lambda = 1/2$	$\lambda = 1/4$
Mate-pair information	w/o mate-pair	w/i mate-pair	w/i mate-pair
Total # polymorphism	179,686	179,686	180,294
# polymorphism reported (including singleton)	165,188	165,188	151,501
True positive rate (all reported)	98.2%	96.0%	94.5%
Percentage of correctly detected sites	90.3%	88.3%	79.4%
Average segment length	5.1	33.9	19.4

detected pairs among all is 88.96%. We also include results for the case of $r = 0.5$ without mate-pair information. In the case of haplotype frequency $r = 0.25$, the performance was still satisfactory considering the coverage and sequencing error rates.

1.3.8 A Simulation of Human Diploid Genome

The performance of the method on genomes of less dense polymorphisms is tested by another simulation. We simulate a situation of a polymorphism rate 0.3%, which can be found in some regions of *Homo sapiens*. The results are shown in Table 1.3.

1.3.9 Length of Haplotype Segment and Two-End Sequencing

The gain of two-end sequencing of variable size fragments and the proposed bridging strategy using mate-pair information can be measured by the average length of haplotype segment. In the simulation of the *Ciona intestinalis* genome, on average

Table 1.4 Error patterns. The polymorphism rate is 1.2% as in *Ciona intestinalis*

Truth	Error type	$\lambda = 1/2$	$\lambda = 1/4$
	Reconstructed	Percentage	
Heterozygote	Heterozygote, one match	1.2	2.8
Heterozygote	Wrong phase	1.7	1.2
	Total	3.0	3.9

each haplotype segment contains 70.36 polymorphic sites while it contains only 45.43 polymorphic sites without mate-pair information. In the case of 0.3% polymorphic rates, on average each haplotype segment contains 33.87 polymorphic sites while it contains only 5.06 polymorphic sites without mate-pair information. This shows that two-end sequencing strategy offers significant haplotype information and the bridging strategy works well.

1.3.10 Error Patterns

We categorize false positive errors in Table 1.4. As we can see, phase errors are rare. Some errors are of partial genotypes. Namely, one base in a genotype is mistaken and the other one is correct.

1.3.11 Confidence Scores for Haplotype Segments

The accuracy assessment of haplotype estimation is exemplified in Fig. 1.12. The estimated haplotype segments can be evaluated by the scores coupled with them. We checked the consistency of observed probability scores versus nominal scores calculated by equation (1.8) and recursion (1.7) in Fig. 1.13. When the confidence score is larger than 0.5, the empirical results are quite consistent with expected ones. When the confidence score is smaller than 0.5, the empirical error rates are slightly lower than the nominal ones. This is due to the fact that we reject some phases in the pairwise comparison step using a threshold of 0.5. We can correct the bias in the range of low probability by a straightforward empirical method. We also check confidence scores of each haplotype segment versus number of polymorphic sites. Most of the errors occurred in short segments with less than four polymorphic sites due to factors such as low coverage and relatively large distance from other polymorphic sites.

1.3.12 Gibbs Sampling Algorithm

A more delicate Gibbs Sampling algorithm was used in [9] to maximize the conditional probability $\Pr(\mathbf{S}|\mathbf{X})$. We sketch the basic idea as follows. As a matter of fact,

- Segment 7 starts at 144 and ends at 201 - size: 58 - confidence score: 0.9960188778

  Template 1: ATACCTCGTTCCGAATGCGAATACCGCTCAATAC
  Template 2: TCCATAATGGTAC – TACACCGGTGTATCTCCGGT
  Template 1 (continue): TGAACCTGTAAACCAACGCGTAGA
  Template 2 (continue): ACTCGAGAAGTCTGTGAATTGTAG

- Segment 8 starts at 202 and ends at 230 - size: 29 - confidence score: 0.9752074893

  Template 1: TAATTACCATAGTGACATCAGTTCAATTT
  Template 2: ACGCCGTACCCTCCTA–AGCAAAACGACA

- Segment 9 starts at 231 and ends at 254 - size: 24 - confidence score: 0.6125626073

  Template 1: ATGCCAACATTCTCCCCGCAGCTA
  Template 2: –GAAATTGTGGTGTTAATTGAGAT

Fig. 1.12 Examples of accuracy assessment. The haplotype segments with positions, sizes, and confidence scores are produced by our program

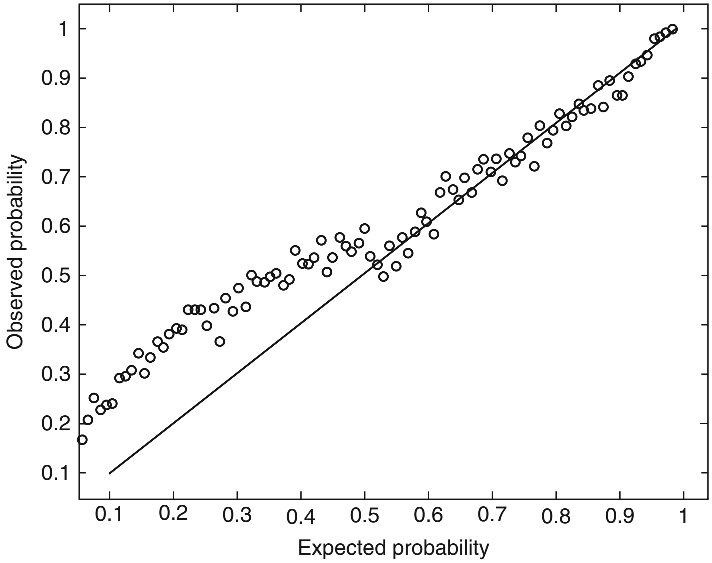


Fig. 1.13 Observed error rates vs. nominal scores calculated by equation (1.8) and recursion (1.7)

the conditional distribution $\Pr(\mathbf{S}|\mathbf{F},\mathbf{X})$ is essentially the one-chromosome problem solved in Churchill and Waterman [3]. We can compute $\Pr(\mathbf{S},\mathbf{F}|\mathbf{X})$ by alternating the following steps:

1. generate $s^{(l+1)}$ from $\Pr(\mathbf{S}|\mathbf{F} = f^{(l)}, \mathbf{X} = x)$;
2. generate $f^{(l+1)}$ from $\Pr(\mathbf{F}|\mathbf{S} = s^{(l+1)}, \mathbf{X} = x)$.

As we mentioned earlier, the distribution of $\Pr(\mathbf{F}|\mathbf{S} = s, \mathbf{X} = x)$ is also difficult to obtain. A remedy is to replace Step 2 by a series of sampling. That is, we update one fragment membership while keeping other fragment memberships unchanged, and carry out the operation through all the fragments.

1.3.13 Diploid Genome of Ciona intestinalis *and Comparative Genomic Studies*

We applied the method to reconstruct the diploid genome using the whole-genome shotgun sequencing data, see http://genome.jgi-psf.org/ciona4/ciona4.download.ftp.html. Namely, we first align reads to the published reference sequence, and then apply Algorithm 1 to each scaffold. Figure 1.14 shows a part of one scaffold, in which nine fragments were aligned; the two targets are shown at the top. Four polymorphic sites including an indel `CCC/---` were observed in this region of about 40 nucleotides. This is quite typical in *Ciona intestinalis* genome [4].

We successfully applied the above probabilistic framework and the Gibbs sampling algorithm to reconstruct the diploid genome of *Ciona intestinalis* from the shotgun sequencing reads obtained from JGI. The new genomic knowledge is achieved without any additional penny in wet lab work. According to our reconstruction, 85.4% of predicted gene sequences are continuously covered by single haplotype segments. We estimate the polymorphism rate of *Ciona intestinalis* to be 1.2 and 1.5%, according to two different polymorphism counting schemes. The result shows that heterozygosity number in a window of 200 bp is well fitted by a geometric Poisson distribution. After the publication of the *Ciona intestinalis* diploid genome [9], the diploid genome sequences of an individual human were reported in [12] using a different method.

We also conducted a comparative analysis with *Ciona savignyi*, and discovered interesting patterns of conserved DNA elements in chordates, see [9]. Most conserved elements found in exons are relatively long, while many highly conserved yet

```
scaffold_1611 #1 aaCgagataatagaatTagaagtgt---atcttcccca-Ccct-t
              #2 aaTgagataatagaatAagaagtgtCCCatcttcccca-Acct-t

                 aaCgagataatagaatTagaagtgt---atcttcccca-Ccct-t
                 aaCgagataatagaatTagaagtgt---atcttcccca-Ccct-t
                 aaCgagataatagaatTagaagtgt---atcttcccca-Ccct-t
                 aaCgagataatagaatTagaagtgt---atcttcccca-Ccct-t
                 aaTgagataatagaatAagaagtgtCCCatcttcccca-Acct-t
                 aaCgagataatagaatTagaagtgt---atcttccccacCcct-t
                 aaTgagataatagaatAagaagtgtCCCatcttcccca-Acct-t
                 -aCgagataatagaatTagaagtgt---atcttcccca-Ccct-t
                               atTagaagtgt---atcttcccca-Ccctgt
```

Fig. 1.14 Part of a scaffold from *Ciona intestinalis*. Nine fragments were aligned and four polymorphic sites were observed. Nonpolymorphic and polymorphic sites are represented by small and large letters respectively. The two targets are shown at the top

relatively short elements are found in intergenic regions. These insights can hardly be obtained without an accurate haplotype estimation.

1.4 Discussion

1.4.1 Alignment Algorithm

We have observed that different alignment algorithms may produce slightly different assembly matrices. Our adjustment of scores is adaptive to alignment in the sense that it optimizes performance based on each assembly. When a new alignment procedure is used, adjustment may change correspondingly.

Phrap (see http://www.phrap.org) examines all individual sequences at a given position, and generally uses the highest quality sequence to build the consensus. *Phrap* also uses the quality information of individual sequences to estimate the quality of the consensus sequence. In comparison, our method can be used with any other assembly algorithms. The reconstruction of consensus and definition of quality value are based on a probabilistic model. It can adjust potential bias of quality scores in base-calling.

1.4.2 Computing Complexity

In the logistic model, the inner loop computes the parameters in logistic regressions by either the Fisher scoring method or by the Newton-Raphson method. Both methods converge quadratically. The outer loop is an E-M procedure, and in general a E-M algorithm converges at a linear rate. Thus the computing complexity hinges on the E-M algorithm. Specifically, let κ, L, D_1, D_2 be the coverage, number of knots, number of the Newton iterations, number of the E-M iterations, respectively, then the complexity is about $O((n\kappa + L^3 D_1)D_2)$, where n is the size of the target DNA. Similarly, the complexity for the conditional sequencing error model is $O(n\kappa D)$, where D is the number of the E-M iterations.

1.4.3 Repeat Patterns

We have checked the errors that are left uncorrected by the procedure described in this chapter. Almost all of them are from regions with repeat patterns. They can be single-nucleotide, di-nucleotide, or tri-nucleotide repeats. Situations become even more subtle if two repeats are next to another. In one example, an A is in the middle of four Cs and is missed. In these cases, it is not appropriate to assume that the sequencing error pattern is independent of local contexts. We are considering more sophisticated models to deal with regions with repeats. Li and Speed [13, 16] proposed a parametric deconvolution procedure to improve accuracy of sequencing for regions with repeats.

1.4.4 Size of Training Data Set

We have applied our method to data sets of different sizes. The larger the data set, the more knots are selected in the optimal model. We can achieve satisfactory training with an assembly of size 30 kb and a coverage of six. The result is not sensitive to the "quality" of quality scores. In comparison, the training of *Phred* scores requires several hundred million base-calls, and it has been carried out on the sequencing traces generated from ABI sequencers. It is difficult to obtain reliable quality scores for other sequencers by the *Phred* training method if only limited base-calling data are available. In this situation, we can apply the method proposed in this chapter to adjust preliminary quality scores obtained under roughly the same condition and get probabilistically meaningful quality scores. Earlier we reported one such example that calibrates Beckman *CEQ* quality scores using 500 kb from an *Arabidopsis* resequencing project.

1.4.5 Next Generation Sequencing

Quite some yet different new sequencing technologies have emerged recently. Although the details of the chemistry, physics and molecular biology vary from one scheme to another, some parts of the mathematical and statistical analysis in the measurement problems more or less remain unchanged. For example, the color correction method exploited in the Solexa sequencing adopts the one we originally developed for Sanger sequencing [15].

The calibration of quality scores and the framework presented in the first half is applicable to the new-generation sequencing systems, particularly to Illumina/Solexa reads, in which miscalls are the primary base call errors. The Solexa system produces four scores corresponding to four bases at one position. In fact, from quality scores our logistic model generates probabilistic calls that include four components, see Fig. 1.7. The method has several advantages. First we can allow mismatches in aligning each read to a target genome. Second, based on the probabilistic model, we can evaluate the chance of each alignment and this may lead to more accurate results. Even though the reads from SBS systems are short, the diploid genome framework described in the second half of the chapter is also applicable to the new generation sequencing, especially with the availability of paired-end reads.

Acknowledgements We thank Prof. Michael Waterman for initiating the works reported in this chapter. This work is supported by the NIH CEGS grant to University of Southern California and the NIH grant R01 GM75308;.

References

1. Adams, M. D., Fields, C., & Ventor, J. C. (Eds.). (1994). *Automated DNA sequencing and analysis*. London, San Diego: Academic.
2. An, H., & Gu, L. (1985). On the selection of regression variables. *Acta Mathematicae Applicatae Sinica*, *2*, 27–36.
3. Churchill, G. A., & Waterman, M. S. (1992). The accuracy of DNA sequences: Estimating sequence quality. *Genomics*, *14*, 89–98.
4. Dehal, P., et al. (2002). The draft genome of *ciona intestinalis*: Insights into chordate and vertebrate origins. *Science*, *298*, 2157–2167.
5. Ewing, B., & Green, P. (1998). Base-calling of automated sequencer traces using *phred*. 2. error probabilities. *Genome Research*, *8*, 186–194.
6. Ewing, B., et al. (1998). Base-calling of automated sequencer traces using *phred*. 1. accuracy assessment. *Genome Research*, *8*, 175–185.
7. Felsenfeld, A., Peterson, J., Schloss, J., & Guyer, M. (1999). Assessing the quality of the DNA sequence from the human genome project. *Genome Research*, *9*, 1–4.
8. Kim, J. H., Waterman, M. S., & Li, L. M. (2006). Accuracy assessment of diploid consensus sequences. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *4*, 88–97.
9. Kim, J. H., Waterman, M. S., & Li, L. M. (2007). Diploid genome reconstruction of *Ciona intestinalis* and comparative analysis with *Ciona savignyi*. *Genome Research*, *17*, 1101–1110.
10. Lancia, G., Bafna, V., Istrail, S., Lippert, R., & Schwartz, R. (2001). SNPs problems, complexity, and algorithms. In *European symposium on algorithms* (pp. 182–193). *Lecture Notes in Computer Science*. Springer-Verlag GmbH.
11. Lander, E. S., & Waterman, M. S. (1988). Genomic mapping by fingerprinting random clones. *Genomics*, *2*, 231–239.
12. Levy, S., et al. (2007). The diploid genome sequence of an individual human. *PLoS Biology*, *5*, e254. dOi:10.1371/journal.pbio.0050254.
13. Li, L. M. (2002). DNA sequencing and parametric deconvolution. *Statistica Sinica*, *12*, 179–202.
14. Li, L. M., Kim, J. H., & Waterman, M. S. (2004). Haplotype reconstruction from SNP alignment. *Journal of Computational Biology*, *11*, 505–516.
15. Li, L. M., & Speed, T. P. (1999). An estimate of the color separation matrix in four-dye fluorescence-based DNA sequencing. *Electrophoresis*, *20*, 1433–1442.
16. Li, L. M., & Speed, T. P. (2002). Parametric deconvolution of positive spike trains. *Annals of Statistics*, *28*, 1279–1301.
17. Lippert, R., Schwartz, R., Lancia, G., & Istrail, S. (2002). Algorithmic strategies for the SNP haplotype assembly problem. *Briefings in Bioinformatics*, *3*, 1–9.
18. McCullagh, P., & Nelder, J. A. (1989). *Generalized linear model* (2nd ed.). London: Chapman and Hall.
19. Nelson, D. O., & Fridlyand, J. (2003). Designing meaningful measures of real length for data produced by DNA sequencers. In *Science and statistics: A festschrift for Terry Speed* (pp. 295–306). *Lecture Notes-Monograph Series*. Institute of Mathematical Statistics.
20. Parkhill, J., et al. (2000). The genome sequence of the food-borne pathogen *campylobacter jejuni* reveals hypervariable sequences. *Nature*, *403*, 665–668.
21. Ross, S. M. (1989). *Introduction to probability models* (4th ed.). Academic.
22. Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461–464.
23. Venables, W. N., & Ripley, B. D. (1994). *Modern applied statistics with S-plus*. Springer.
24. Winer, R., Yen, G., & Huang, J. (2002). *Call scores and quality values: Two measures of quality produced by the CEQ® genetic analysis systems*. Beckman Coulter, Inc.

Chapter 2
Statistical and Computational Studies on Alternative Splicing

Liang Chen

Abstract The accumulating genome sequences and other high-throughput data have shed light on the extent and importance of alternative splicing in functional regulation. Alternative splicing dramatically increases the transcriptome and proteome diversity of higher organisms by producing multiple splice variants from different combinations of exons. It has an important role in many biological processes including nervous system development and programmed cell death. Many human diseases including cancer arise from defects in alternative splicing and its regulation. This chapter reviews statistical and computational methods on genome-wide alternative splicing studies.

2.1 Introduction

Alternative pre-mRNA splicing is a prevalent post-transcriptional gene regulation mechanism which has been estimated to occur in more than 90% of human genes [1,2]. During alternative splicing, multiple transcript isoforms produced from a single gene can lead to protein isoforms with distinct functions, which greatly expands proteomic diversity in higher eukaryotes. The alternative splicing of multiple pre-mRNAs is tightly regulated and coordinated, and is an essential component for many biological processes including nervous system development and programmed cell death. The phenomenon of alternative splicing was first discovered in concept in 1978 [3], and was then verified experimentally in 1987 [4]. Alternative splicing was previously thought as a relatively uncommon form of gene regulation. With the accumulation of Expressed Sequence Tags (EST) and mRNA data sets, genome-wide studies on alternative splicing demonstrated that as many as 60% of the human genes were alternatively spliced [5–8]. The percentage was further

L. Chen
Molecular and Computational Biology, Department of Biological Sciences, University of Southern California, 1050 Childs Way, Los Angeles, California 90089, USA
e-mail: liang.chen@usc.edu

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_2,

© Springer-Verlag Berlin Heidelberg 2011

increased to 90% which was estimated by the most recent high-throughput sequencing technology [1, 2]. In addition, there is striking variation in alternative splicing across different tissues or different developmental stages [5]. These results indicate that alternative splicing plays an important role in increasing functional complexity in higher organisms rather than the exception in gene expression. With the availability of multiple genome sequences and high-throughput techniques, it is feasible to study alternative splicing on a genomic scale. Here we present an overview of the statistical and computational studies on alternative splicing, and important findings and challenges are highlighted and discussed.

2.2 Types of Alternative Splicing

Alternative splicing events can be classified into cassette exon, mutually exclusive exons, retained intron, alternative 5′ splice sites, alternative 3′ splice sites, alternative promoters, and alternative poly-A sites (Fig. 2.1). The most common type of alternative splicing is including or skipping a cassette exon in the mature mRNA. A pair of exons can be mutually exclusively spliced with only one exon included in the mature mRNA but not both. The excision of an intron can be suppressed, which results in the retention of the entire intron. And exons can be extended or shortened through the use of alternative 5′ or 3′ splice sites. Strictly speaking, alternative promoters and alternative poly-A sites are alternative selection of transcription start sites or poly-A sites and are not due to alternative splicing per se. Among these

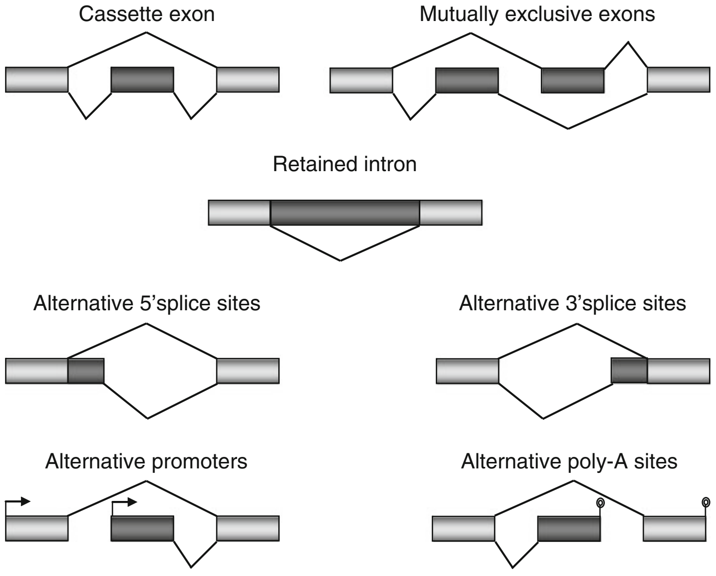


Fig. 2.1 Types of alternative splicing events

alternative splicing events, intron retention is generally the most difficult type to detect because it is hard to distinguish from experimental artifacts. For instance, incompletely spliced transcripts contain intron fragments, which could be mistakenly considered as intron retention. Many genes have multiple alternative splicing events with complex combinations of exons, producing a family of diverse transcript isoforms. For example, in Drosophila melanogaster, gene Dscam can potentially produce 38,016 different mature mRNAs by different combinations of 95 cassette exons [9–11].

2.3 Global Identification of Alternative Splicing Events

2.3.1 Identifying Alternative Splicing by Sequence Alignment

One way to identify alternative splicing events is based on the alignment of ESTs with genomic and mRNA sequences. EST sequences are short fragments of transcribed cDNA sequences, usually 300–400 base pair (bp). They are produced by shotgun sequencing of one or both strands of a cloned mRNA. About 61 million ESTs have been deposited in the public dbEST database (dated as April, 2009, all species). A number of programs have been developed to align ESTs against the complete genome sequences efficiently. For example, BLAT is a “BLAST-Like Alignment Tool” which uses a hashing and indexing algorithm [12]. It is about 500 times faster than BLAST for mRNA/DNA alignments. Given the alignments of ESTs and genomic sequences, we can mark the locations of exons and introns. The comparisons of exon-intron structures further distinguish the alternative splicing events. Sometimes, an EST can be mapped to multiple genomic positions with high alignment scores. These genome alignments can be further corrected by considering consensus splice sites. For example, alignment tools SIM4 [13], GMAP [14], and SPA [15] consider GT…AG consensus splice sites to generate valid alignments. Although the sequence alignment approaches have made much progress in alternative splicing detection, challenges remain in dealing with non-canonical splice junctions, detection of small exons, high EST sequencing errors, bias inherent to EST preparation, and so on. Other limitations include the insufficient sequence coverage for some transcripts and the biased sampling to a limited number of cell and tissue types.

After the identification of individual alternative splicing events, a more complicated task is the construction of full-length alternatively spliced transcripts. “Splice graph” has been introduced to facilitate the construction of full-length transcript isoforms [16–19]. The splice graph represents a gene as a directed acyclic graph in which exons are represented as vertices and each splice junction is represented as a directed edge between two exons (see example in Fig. 2.2). Splice variants can be inferred by graph algorithms to traverse the graph from a start vertex with no incoming arcs to an end vertex with no outgoing arcs. A large number of potential splice variants can be enumerated from a splice graph, but many of them may be

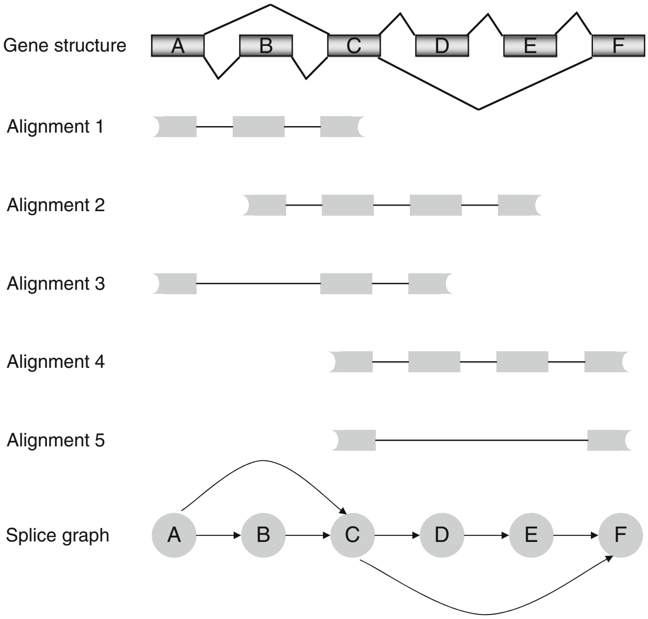


Fig. 2.2 Splice graph constructed from EST alignments to reference genome. The underlying true gene structure and the observed evidence alignments are also shown

artificial constructs without biological relevance because exons are not randomly joined to produce all possible transcript isoforms. Several methods have been proposed to select or prioritize candidate transcripts which are most likely to exist given the sequence observations. For example, AIR is an integrated software system for gene and alternative splicing annotation [16]. It assigns different scores to different splicing variants based on its support by evidence such as mapping quality, the length of alignment, accuracy of splice signals, and the level of fragmentation of evidence alignments. High-scoring splice variants were further selected for the annotation. ECgene algorithm assesses each possible splice variant based on the sequence quality and the number of cDNA alignments [18]. Xing et al. applied the Expectation-Maximization algorithm to identify the most likely traversals based on the observed number of alignments along the gene [19]. The performance of these methods is limited by the contamination of ESTs with genomic fragments, alignment errors, and so on.

2.3.2 Identifying Alternative Splicing by Sequence Content and Conservation

Because mRNA alternative splicing is a highly regulated process, comparative genomics can provide us clues about whether there is an alternative exon in sites with high selection pressure. Alternative methods have been proposed to predict alternatively spliced exons based on machine learning algorithms incorporating features such as sequence content and sequence conservation. Leparc et al. used splice-site sequence Markov models and a Bayesian classifier to identify cassette exons from intron sequences [20]. With additional information from sequence conservation and phosphorylation or protein-binding motifs, they successfully predicted and experimentally confirmed 26 novel human cassette exons which are involved in intracellular signaling. Sorek et al. assembled 243 alternative and 1,753 constitutive exons that are conserved between human and mouse [21, 22]. They identified several features differentiating between alternatively spliced and constitutively spliced exons. Specifically, alternative exons tend to be smaller, have length that is a multiple of 3 (to preserve the protein reading frame), have higher sequence identity between human and mouse sequences, and have higher conservation in the flanking intronic regions. The most important features are the ones based on the sequence similarity between human and mouse. Yeo et al. used sequence features to distinguish alternative splicing events conserved in human and mouse [23]. Chen et al. used the Random Forests algorithm to predict skipped exons using features like position-specific conservation scores [24]. The training data was based on the high-quality annotation of the Encyclopedia of DNA Elements (ENCODE) regions. The pilot project of the ENCODE has rigorously identified functional elements in the 1% region of the human genome. The GENCODE consortium of the ENCODE project has manually prepared a high-quality annotation for transcripts in the ENCODE regions. Chen et al. assembled the lists of skipped exons, constitutive exons and introns as training sets. Using the Random Forest algorithm [25], they were able to identify skipped exons based on the sequence content and conservation features [24]. The Random Forests consist of many decision trees and each tree is constructed by a bootstrap sample from the original data. A decision tree can be treated as a set of Boolean functions of features and these conjunctions of features partition training samples into groups with homogenous class labels. The output of the Random Forests for each test sample is the class with majority votes from these trees. The Random Forests generates an internal unbiased estimate of classification error based on the out-of-bag data during the Forests building process. There is no need for cross-validation or a separate test data.

As shown in Fig. 2.3, there are dramatic differences in the conservation scores of the flanking regions of alternative exons and constitutive exons. Alternative exons have higher conservation level in the flanking intronic regions compared to constitutive exons. These more conserved regions provide good candidates for functional regulatory motifs. The enriched sequence motifs in these regions may participate in the alternative splicing modulation which could be different from the regular splicing process.

Besides the flanking intronic regions, the exonic regions are also involved in the splicing regulation. However, the comparative genomics studies on exonic regions are more complicated, because additional selective pressure is imposed on the coding sequence in order to preserve the protein sequence. It has been shown that the evolution rate is lower for exon regions near the intron-exon boundaries than the

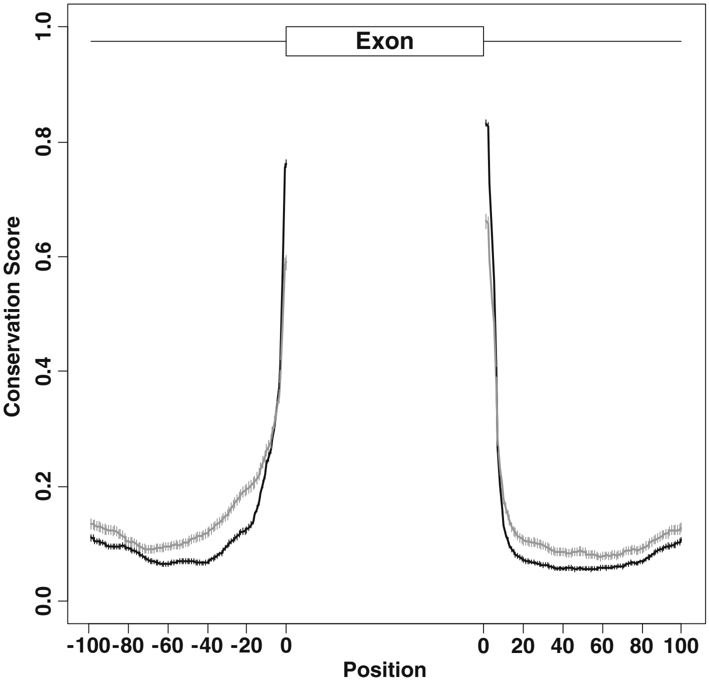


Fig. 2.3 Position-specific conservation for the flanking intronic regions of constitutive exons (*black*) and alternative exons (*grey*) (Adapted from [24]). Y axis is the average conservation score at each position. The error bar indicates the standard error of the mean. Constitutive exons and alternative exons were assembled from the high-quality annotation of the ENCODE project. The conservation score is the PhastCon score from the UCSC Genome Browser (http://genome.ucsc.edu/)

middle part of exons, by estimating the non-synonymous substitution rate and the synonymous substitution rate from the alignment of human-mouse sequences [26]. The SNP density is the lowest near the splice sites, which also indicates that exon regions near the splice sites are under higher selection pressure [27]. These findings suggest that the exon regions near the junctions are involved in splicing regulation. Further studies are needed to distinguish the selection pressure on alternative exons, constitutive exons, and amino acid constrains.

2.3.3 Identify Alternative Splicing by Microarray

Although the sequence alignment and the comparative genomics approaches have made much progress in the prediction of alternative splicing events, they give us only a qualitative rather than a quantitative view of alternative splicing. They only provide evidence about the existence of an alternative splicing event, but cannot give information about its temporal and spatial regulation nor the degree of alternative splicing.

The highly parallel nature of microarray platforms makes it possible to identify and quantify all of alternative splicing for a specific tissue, developmental stage, or disease versus normal conditions of the cell. Traditional microarrays are spotted with EST-derived cDNAs or $3'$-clustered oligonucleotide sequences representing the total transcript abundance. These microarrays are not suitable for alternative splicing studies and special probes need to be designed instead. For example, splice junction arrays bear probes spanning annotated exon-exon junctions for individual splice variant. Johnson et al. designed a set of five Agilent microarrays containing $\sim$125,000 different 36-nucleotide (nt) junction probes to monitor the exon-exon junctions of 10,000 multi-exon Human RefSeq genes across 52 tissues and cell lines [5]. Boutz et al. used splice junction arrays to monitor the reprogrammed alternative splicing during neuronal development [28]. Besides splice junction arrays, alternative arrays uses "exon-centric" probes. For instance, in the design of Affymetrix exon arrays, gene annotations from databases were assembled to infer transcript clusters and exon clusters. A transcript cluster roughly corresponds to a gene. In many cases, an exon cluster represents a true biological exon and it acts as one probe selection region. In other cases, an exon cluster represents the union of multiple overlapping exons possibly due to alternative splice sites. Such exon clusters were further fragmented into multiple probe selection regions according to the hard edges (e.g., splice sites). Multiple probes were designed for each probe selection region as a probe set. The Affymetrix human exon array (1.0 ST) contains approximately 1.4 million probe sets interrogating over one million exon clusters. Analysis of alternative splicing in 16 human tissues with these arrays identified a large number of tissue-specific exons [29]. Yeo et al. used Affymetrix exon arrays to identify the differential alternative splicing between human embryonic stem cells and neural progenitor cells [30]. More recent microarrays include both junction probes and exon body probes. Castle et al. designed probes targeting on exons or junctions to monitor 203,672 exons and 178,351 exon-exon junctions in 17,939 human genes across 48 diverse human tissues and cell lines [31]. In addition, tiled oligonucleotide arrays spanning whole chromosomes or genomes provide comprehensive coverage and avoid the need of prior information about exons. However, this approach is expensive and needs extremely large number of probes. These microarray designs are summarized in Fig. 2.4. In principle, all data analysis tools developed for standard gene microarrays can be used in the analysis of alternative splicing microarrays. The special challenge is how to distinguish splicing signal from transcription signal. The methods outlined below present some tools that have been used on the alternative splicing microarray data analysis.

2.3.3.1 Splicing Index

For the alternative splicing microarray analysis, the most straightforward approach is the splicing index calculation [32]. In the splicing index approach, exon inclusion

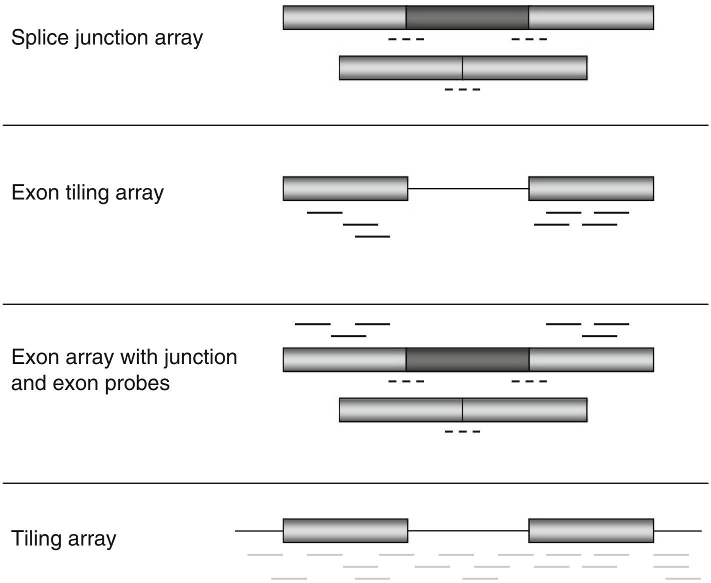


Fig. 2.4 Alternative splicing microarrays. *Black dot lines* represent junction probes. *Black solid lines* represent exon probes. For tiling arrays, probes are designed along the genome disregarding gene structure (*grey lines*)

rates under two conditions are compared to identify differential alternative splicing events. Gene-level normalized exon intensity is defined as the ratio of the exon intensity to the gene intensity. For example, the normalized intensity (NI) for exon i in experiment j is:

$$NI_{ij} = E_{ij}/G_j \tag{2.1}$$

where E_{ij} is the estimated intensity level for exon i in experiment j and G_j is the estimated gene intensity. "Gene intensity" here represents the overall transcript abundance of a gene which may include a family of transcript isoforms. "Gene intensity" can be estimated by dynamic weighting of the most informative probes. It is robust to outliers due to alternative splicing. Thus, the contributions from alternative exons to "gene intensity" are trivial.

A significant difference in the normalized exon intensity indicates that this exon has different inclusion or exclusion rates (relative to the gene level) between two conditions. The splicing index for experiment 1 and experiment 2 is defined as:

$$\text{Splicing index} = log_2(NI_{i1}/NI_{i2}). \tag{2.2}$$

Therefore, an extreme value of splicing index indicates a differential alternative splicing event.

2.3.3.2 ANOSVA

Analysis of splice variation (ANOSVA) uses a statistical testing principle to detect putative splicing variation from expression data [33]. It is based on a two-way analysis of variance (ANOVA) model:

$$y_{ijkl} = \mu + \alpha_i + \beta_j + \gamma_{ij} + error, \tag{2.3}$$

where y_{ijkl} is the observed log intensity of probe k of probe set i (or exon i), measured in experiment j of experiment set l; μ is the baseline intensity level for all probes in all experiments; α_i is the average probe affinity of probe set i; β_j is the experiment effect; and γ_{ij} is the interaction term for probe set i and experiment j. A large change in splicing will result in a large interaction term γ_{ij}. However, due to the limited number of replicates for exon-array experiments and the resultant limited statistical power, it is difficult to identify interactions. Meanwhile, a significant interaction term does not necessarily mean a large change in splicing, because the unfitness of the single-concentration model without the interaction term may be simply due to the high noise level. Preliminary evaluation of ANOSVA on exon array data did not yield good performance (Alternative Transcript Analysis Methods for Exon Arrays Whitepaper, Affymetrix). Therefore, ANOSVA should be used with caution.

2.3.3.3 FIRMA

Instead of estimating the interaction term γ_{ij} explicitly, FIRMA (Finding isoforms using robust multichip analysis) [34] frames the problem of detecting alternative splicing as a problem of outlier detection. In FIRMA, y_{ijk} represents log intensity of probe k of exon i measured in experiment j (signal has been background-corrected and normalized). It is modeled as:

$$y_{ijk} = c_j + p_k + error, \tag{2.4}$$

where c_j is the experiment effect and p_k is the probe effect. The residual from the fitted model is:

$$r_{ijk} = y_{ijk} - \hat{c}_j + \hat{p}_k. \tag{2.5}$$

The residual describes the discrepancy of probe intensity in a given experiment from the expected expression and gives a measure of the hidden interaction term γ_{ij}. The final score statistic is:

$$F_{ij} = median_{k \in exon\ j}(r_{ijk}/s). \tag{2.6}$$

The standard error, s, is calculated by the median absolute deviation (MAD) of the residuals. Compared with ANOSVA, FIRMA can detect alternative splicing without replicates. And the interaction term is not directly inferred and reflected by a robust measure of the residuals instead. FIRMA assumes that the interaction term has limited effect so that c and p are still well estimated in the model without the interaction term.

2.3.3.4 DECONV

The above methods target on each individual exon to determine whether it is differentially spliced or not. They do not require the whole exon-intron structure of a gene. A gene may have multiple positions of alternative splicing and the resulted multiple (>2) splice variants can coexist in the same condition. Another challenging task is to estimating the relative abundance of each variant in one condition. Wang et al. developed a gene structure-based splice variant deconvolution method (DECONV) to estimate the splice variant's concentration [35]. DECONV assumes that there is linear relationship between the probe intensity and the target transcript concentration as proposed by Li and Wong [36]. In the reduced model of Li and Wong,

$$y_{ij} = PM_{ij} - MM_{ij} = a_i x_j + \varepsilon_{ij}, \tag{2.7}$$

where y_{ij} is the intensity level for probe i in experiment j, a_i is the probe affinity, and x_j is the target transcript concentration. DECONV extends the model for multiple splice variants case:

$$\mathbf{Y} = \mathbf{A} \cdot \mathbf{G} \cdot \mathbf{T} + \mathbf{E}, \tag{2.8}$$

where $\mathbf{Y}$ is an I by J matrix with y_{ij} representing the intensity for probe i in experiment j, $\mathbf{A} = diag(a_{11}, \ldots, a_{II})$ is the diagonal matrix of unknown affinities for all of the probes included in the gene; matrix $\mathbf{T} = \{T_{kj}\}$ represents the unknown concentration of the k-th splice variant in the j-th experiment; the property matrix $\mathbf{G} = \{g_{ik}\}$ relates probes with different splice variants according to whether the probe belongs to the transcript or not.

$$\begin{aligned} g_{ik} &= 1 \text{ if probe i belongs to splice variant k,} \\ &= 0 \text{ if probe i does not belong to splice variant k.} \end{aligned} \tag{2.9}$$

And $\mathbf{E}$ is the error term. To estimate the unknown $\mathbf{A}$ and $\mathbf{T}$, they minimize the function:

$$f(\mathbf{A}, \mathbf{T}) = (\|\mathbf{Y} - \mathbf{AGT}\|_2)^2, \tag{2.10}$$

under the constraints:

$$\sum_{i=1}^{I} a_{ii}^2 = constant, \tag{2.11}$$

$$a_{ii} \geq 0, \tag{2.12}$$

$$t_{kj} \geq 0. \tag{2.13}$$

The maximum likelihood estimation framework is finally used by iteratively fixing **A** and solving for **T**, then fixing **T** and solving for **A** until convergence. DECONV works well for genes with two transcript isoforms, but is less than perfect for genes with three or more isoforms. DECONV requires the complete information about the number and the structure of all possible splice variants for a gene. It is not intended for the discovery of new splice variants.

2.3.3.5 SPACE

A similar algorithm, SPACE (splicing prediction and concentration estimation), was proposed to predict the structures and the abundances of transcript isoforms from microarray data [37]. Besides matrices **A** and **T**, they also treated the gene structure matrix **G** as unknown. A "non-negative matrix factorization" method was applied to handle the non-negative constraints and factorize **Y** into **W** and **H**:

$$\mathbf{Y_{IJ}} \approx \mathbf{W_{IK}} \cdot \mathbf{H_{KJ}}. \tag{2.14}$$

Remember that $\mathbf{Y} \approx \mathbf{A} \cdot \mathbf{G} \cdot \mathbf{T}$, so **H** gives the relative concentration of each splice variant and **W** contains information of both probe affinity and gene structure. Specifically, they used the maximum value of each row of the **W** matrix as the affinity of the corresponding probe.

$$\begin{aligned} a_{ii} &= max_k(W_{ik}) \\ \mathbf{G} &= \mathbf{A^{-1}W}. \end{aligned} \tag{2.15}$$

Here **G** will be a matrix whose entries are between 0 and 1. There is a slight change in the definition of **G**:

$$\begin{aligned} g_{ik} &= 1 \;\; \text{if probe i belongs to splice variant k,} \\ &= 0 \;\; \text{if probe i does not belong to splice variant k,} \\ &= \alpha \;\; \text{if probe i partially hybridizes with splice variant k.} \end{aligned} \tag{2.16}$$

The authors reported that the estimation of isoform structure and abundance depends on the number of experiments. When there are only a few experiments (e.g., 5), the estimation error tends to be high. They also mentioned that the model works better if the array includes more probes that are able to distinguish different isoforms or if several different experimental conditions with high variability are considered.

2.3.3.6 GenASAP

Shai et al. developed the GenASAP (Generative model for the alternative splicing array platform) algorithm to infer the expression levels of transcript isoforms

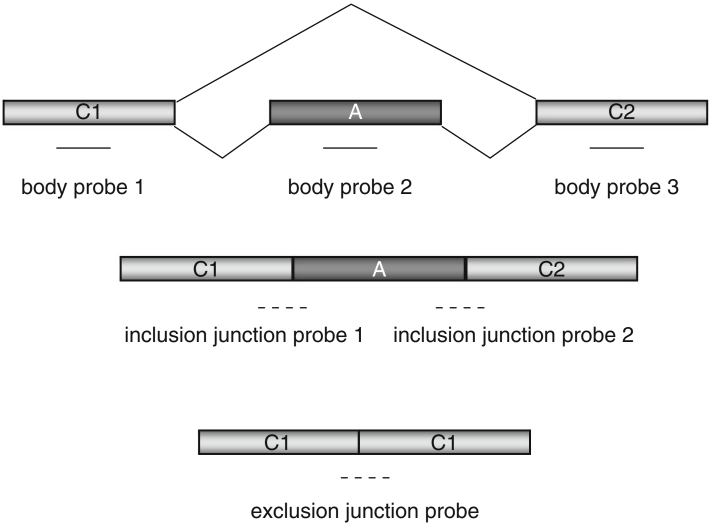


Fig. 2.5 Custom microarray design for cassette exons. *Dot lines* represent junction probes. *Solid lines* represent exon body probes

including or excluding a cassette exon [38]. This was designed specifically for a custom microarray in which an exon-skipping event is represented by three exon body probes and three junction probes (see Fig. 2.5). The probe intensity x_i can be written as:

$$x_i = \lambda_{i1}s_1 + \lambda_{i2}s_2 + \varepsilon_i, \tag{2.17}$$

where x_i is one of the six intensity values for the six specially designed probes, s_1 and s_2 are the two unknown concentrations of the transcript isoforms, λ_{i1} and λ_{i2} are the affinity between probe i and the two transcript isoforms, and ε_i is the error term. To account for the scale-dependent noise and the outliers, the above model is changed to:

$$x_i = (r(\lambda_{i1}s_1 + \lambda_{i2}s_2 + \varepsilon_i))^{1-o_i}(\zeta_i)^{o_i}, \tag{2.18}$$

where r is the scale factor accounting for noise levels at the measured intensity, ζ_i is a pure noise component for the outlier, and o_i is the binary indicator whether the probe measurement is an outlier or not. The conditional probability can be written as:

$$P(\mathbf{X}|\mathbf{S}, r, \mathbf{O}) = \prod_i \mathcal{N}(x_i; r(\lambda_{i1}s_1 + \lambda_{i2}s_2), r^2\psi_i)^{1-o_i}\mathcal{N}(x_i; \varepsilon_i, \nu_i)^{o_i}, \tag{2.19}$$

where $\mathcal{N}(x; \mu, \sigma^2)$ indicates the density of point x under normal distribution with mean μ and variance σ^2. The variance of probe intensity is $r^2\psi_i$. The mean and variance for outliers are ε_i and ν_i. And it assumes independence among probes. The authors used a truncated normal distribution ($\beta \geq 0$) to satisfy the non-negative constraint on isoform abundance and maximized the lower bound of the log likelihood instead of the log likelihood itself during their variational EM learning because the exact posterior cannot be computed.

GenASAP performs well on the abundance estimation and outperforms many supervised methods. It has been successfully applied to the analysis of alternative splicing in mammalian cells and tissues [39, 40]. But it is specific to the focused probe design. In addition, if a gene has more than one alternative exon and more than two transcript isoforms consequently, GenASAP cannot distinguish isoforms which all include the tested cassette exon, neither can it further distinguish isoforms which all exclude the tested cassette exon.

2.3.4 Identify Alternative Splicing by High Throughput Sequencing

Recently, high-throughput sequencing based approach (RNA-Seq) has also been developed to map and quantify transcriptomes. Poly(A)+ mRNAs are purified from cells and fragmented to small size (e.g., ~200 bp). Then they are converted into cDNA and sequenced by the high-throughput sequencing techniques. Sequence tags or reads (usually about $25 \sim 50$ bp for Solexa and SOLid or $250 \sim 400$ bp for 454, and the length expected to increase slightly) from the sequencing machines are mapped to genes and used as a quantitative measure of the expression level. RNA-seq has been successfully applied to yeast [41,42], Arabidopsis thaliana [43], mouse [44, 45], and human [1, 2, 46, 47]. For RNA-seq data, inclusion or exclusion rate of an exon was calculated based on the exon body reads, the flanking inclusion junction reads, and the exclusion junction reads. For example, Wang et al. used the "percent spliced in"(PSI or Ψ) values to determine the fraction of mRNA containing an exon [1]. The PSI value was estimated as the ratio of the density of inclusion reads (i.e. reads per position in regions supporting the inclusion isoform) to the sum of the densities of inclusion and exclusion reads. Pan et al. used the inclusion and exclusion junction reads to quantify the transcript percentage [2]. In their study, the results from RNA-seq data are consistent with results which are from custom microarrays mentioned in GeneASAP. The correlation is 0.8 when applying a threshold of 20 or more reads in one experiment that match at least one of the three splice junctions representing inclusion or skipping of a cassette exon. The correlation increases to 0.85 when a threshold of 50 or more junction reads is applied.

Besides the analysis at the individual exon level, Jiang and Wong developed a method to estimate the transcript isoform abundance from RNA-seq data [48]. This is achieved by solving a Poisson model. Suppose a gene has m exons with lengths $\mathbf{L} = (l_1, \ldots, l_m)$ and n transcript isoforms with expressions $\Theta = (\theta_1, \ldots, \theta_n)$. If two isoforms share part of an exon, the exon was split into several parts and each part was treated as an exon respectively. The count of reads falling a specific region s (e.g., an exon or an exon-exon junction) is the observed data X_s. Let w be the total number of mapped reads. Then X follows a Poisson distribution with mean λ.

When s is exon j, $\lambda = l_j w \sum_{i=1}^{n} c_{ij}\theta_i$ where c_{ij} is 1 if isoform i contains exon j and 0 otherwise. When s is an exon-exon junction, $\lambda = lw \sum_{i=1}^{n} c_{ij}c_{ik}\theta_i$ where l is the length of the junction region, and j and k are indices of the two exons involved in the junction. Assuming the independence among different regions, the joint log-likelihood function can be written as:

$$log(\mathscr{L}(\Theta|x_s, s \in S)) = \sum_{s \in S} log(\mathscr{L}(\Theta|x_s)). \tag{2.20}$$

The isoform abundance θ's can be obtained by the maximum likelihood estimate (MLE). When the true isoform abundance θ is not on the boundary of the parameter space, the distribution of $\hat{\Theta}$ can be approximated asymptotically by a normal distribution with mean Θ and covariance matrix equal to the inverse Fisher information matrix $I(\Theta)^{-1}$. However, in one experimental condition, many isoforms are lowly expressed and the likelihood function is truncated at $\theta_i = 0$. The constraints $\theta_i \geq 0$ for all i make the covariance matrix estimated by $I(\Theta)^{-1}$ unreliable. Instead, they developed a Bayesian inference method based on importance sampling form the posterior distribution of θ's. They utilized the RefSeq mouse annotations and applied their model to a RNA-seq data set. Their results have good consistency with RT-PCR experiments (Pearson's correlation coefficient >0.6).

Instead of estimating the isoform abundance in each experiment, Zheng and Chen proposed a hierarchical Bayesian model, BASIS (Bayesian analysis of splicing isoforms), to identify differentially expressed transcript isoforms between two experiments. BASIS can be applied to both tiling array data and RNA-seq data [49]. For each probe i that appears in at least one transcript isoform of gene g, consider the linear model:

$$\Delta y_{gi} = \sum \Delta\beta_{gj} x_{gij} + \Delta\varepsilon_{gi}, \tag{2.21}$$

where Δy_{gi} is the intensity difference between two conditions for probe i of gene g ($\Delta y_{gi} = y_{gi}^1 - y_{gi}^2$, the intensity is background corrected and normalized), $\Delta\beta_{gj}$ is the expression difference between two conditions for the j-th transcript isoform of gene g, x_{gij} is the binary indicator of whether probe i belongs to isoform j's exon region, and $\Delta\varepsilon_{gi}$ is the error term. Within one data set, g ranges from 1 to G, where G is the total number of genes; i ranges from 1 to n_g where n_g is the total number of probes for gene g; and j ranges from 1 to s_g where s_g is the total number of transcript isoforms for gene g. The total $\Delta\varepsilon_{gi}$'s ($g = 1, \ldots, G$ and $i = 1, \ldots, n_g$) are divided into 100 bins. Each bin contains thousands of probes with similar values. Because probe intensity variance is dependent on probe intensity mean, probes in the same bin exhibit similar variances. The same model can be specified for RNA-seq data with y representing the read coverage over each position.

A hierarchical Bayesian model is constructed as:

$$\Delta\mathbf{Y}_g | \Delta\boldsymbol{\beta}_g, \boldsymbol{\Sigma}_g \sim \mathscr{N}_{n_g}(\mathbf{X}_g \Delta\boldsymbol{\beta}_g, \boldsymbol{\Sigma}_g), \ g = 1, \ldots, G;$$

$$\boldsymbol{\Sigma}_g \equiv diag(\pi_{g1}, \ldots, \pi_{gn_g}), \pi_{gi} = \delta_m \text{ if probe (or position) } i \text{ of gene } g \in \text{bin } m;$$

$$\begin{aligned} \delta_m &\sim IG(\nu/2, \nu\lambda/2),\ m = 1, \ldots, 100; \\ \Delta\boldsymbol{\beta}_g|\boldsymbol{\gamma}_g &\sim \mathscr{N}_{s_g}(0, \mathbf{R}_g); \\ \mathbf{R}_g \equiv diag(\kappa_{g1}, \ldots, \kappa_{gs_g}), \kappa_{gj} &= \tau_{gj} \text{ if } \gamma_{gj} = 0 \text{ and } \kappa_{gj} = \psi_{gj} \text{ if } \gamma_{gj} = 1; \\ f(\boldsymbol{\gamma}_g) &= \prod_{j=1}^{s_g} p^{\gamma_{gj}}(1-p)^{1-\gamma_{gj}}; \end{aligned}$$

where $\Delta\mathbf{Y}_g$, $\Delta\boldsymbol{\beta}_g$, and $\mathbf{X}_g$ are matrices with elements described before, $\boldsymbol{\gamma}_g$ is a latent variable, $\mathscr{N}_{n_g}$ and $\mathscr{N}_{s_g}$ stand for multivariate normal distributions, and IG stands for the inverse gamma distribution. Given the isoform amount differences ($\Delta\boldsymbol{\beta}_g$) and the probe arrangements ($\mathbf{X}_g$), the probe intensity (or read coverage) differences ($\Delta\mathbf{Y}_g$) follow a multivariate normal distribution with mean $\mathbf{X}_g\Delta\boldsymbol{\beta}_g$ and variance $\boldsymbol{\Sigma}_g$. For the variance $\boldsymbol{\Sigma}_g$, specifically, if a probe (or position) is assigned to bin m, the variance of the intensity (or coverage) difference is δ_m. δ_m itself is a random variable following an inverse gamma distribution. γ_{gj} is an indicator whether the j-th isoform is differentially expressed. When $\gamma_{gj} = 0$, the isoform difference $\Delta\beta_{gj} \sim \mathscr{N}(0, \tau_{gj})$ and when $\gamma_{gj} = 1$, $\Delta\beta_{gj} \sim \mathscr{N}(0, \psi_{gj})$. Here $\mathscr{N}$ stands for normal distribution. τ_{gj} was set as a small value so that when $\gamma_{gj} = 0$, $\Delta\beta_{gj}$ is small enough to be estimated as 0. ψ_{gj} was set as a large value so that when $\gamma_{gj} = 1$,$\Delta\beta_{gj}$ is large enough to be included in the final model. Therefore, the latent variable γ can perform variable selection for the linear model. The errors for probes belonging to the same gene can be heteroscedastic and assigned to different bins. In the prior distributions for parameters $\Delta\boldsymbol{\beta}, \boldsymbol{\delta}, \boldsymbol{\gamma}$, there are hyperparameters (τ, ψ, ν, p). Model parameters were inferred based on an ergodic Markov chain generated by the Gibbs sampler.

In summary, a latent variable was introduced to perform direct statistical selection of differentially expressed isoforms. BASIS has the ability to borrow information across different probes (or positions) from the same genes and different genes. It can handle the heteroscedasticity of probe intensity or sequence read coverage, and has been successfully applied to a whole-genome human tiling array data and a mouse RNA-seq data. The authors also found that the power of BASIS is related to gene structure [49]. Specifically, if a gene has more probes (or positions), the power of BASIS is larger. If the difference among isoforms is larger, the power of BASIS is larger. BASIS does not rely on the percentage of isoform-specific positions, and it considers the joint behavior of positions. The model also depends on the completeness of the known splicing patterns of each gene. The authors utilized the Alternative Splicing and Transcript Diversity database [50]. As information accumulates and novel transcript isoforms are discovered, a more accurate and complete alternative splicing annotation database will further improve results derived from BASIS.

2.4 Alternative Splicing Regulation in Eukaryotes

The splicing of pre-mRNA transcripts is carried out by spliceosomes which are large ribonucleoprotein complexes with more than 100 core proteins and five small nuclear RNAs [51, 52]. Besides the core splicing factors, there are additional trans-acting splicing regulators. Consequently, in addition to the core splicing signals including the 5′ splice site (5′ ss), the 3′ splice site (3′ ss), and the branch point sequence (BPS), there is a large amount of splicing regulatory elements for both constitutive exons and alternative exons. These splicing regulatory elements (SREs) can be further classified as exonic splicing enhancers (ESEs), exonic splicing silencer (ESSs), intronic splicing enhancers (ISEs), or intronic splicing silencers (ISSs) based on their locations and functions. Due to the selective constraints, enhancers are expected to play predominant roles in the efficient constitutive splicing, and silencers are expected to play predominant roles in the control of alternative splicing [53]. Large-scale screens of exonic SREs have been conducted experimentally and computationally. Fewer screens for intronic SREs were performed although intronic SREs may have a more prominent role in the alternative splicing regulation because the intronic regions flanking alternative exons are more conserved than those flanking constitutive exons. Motif discovery methods commonly used in transcription factor binding motif identification, in principle, can also be used for the splicing regulatory motif finding. Compared with transcription factor binding sites, the SREs are usually shorter, more degenerate, and have less information content. This poses additional challenges to predict SREs. Similar as the DNA motifs for transcription factor binding, multiple copies of SREs for a single exon will increase their effect on splicing regulation [54–58]. Experimental approaches like cross-linking/immunoprecipitation (CLIP), RNP immunoprecipitation (RIP), and genomic SELEX were applied to identify the binding sites of RNA-binding proteins. Those approaches can be further extended to genome-wide studies of SREs. However, similar as transcription factors, the binding of splicing regulators may not necessarily lead to the regulation.

In the process of alternative splicing, splicing regulators bind to various pre-mRNAs and affect a large number of exons. Meanwhile the splicing pattern of a specific exon is determined by multiple pre-mRNA-binding proteins [59,60]. Therefore, it is particularly interesting and challenging to study how the splicing of a group of exons is co-regulated; how the splicing of an exon is combinatorially controlled by multiple regulators; and what are the general rules of "splicing code" (a set of rules that can predict the splicing patterns of pre-mRNAs [60, 61]). In a recent study of alternative splicing across tissues, association links between genes and exons were identified through partial correlation studies [62]. This method was named pCastNet (partial Correlation analysis of splicing transcriptome Network). These association links can provide information about the regulation relationship between genes and the splicing of exons. It will help us to understand the gene regulation at an exon-level resolution.

We first introduce some notations. If the Pearson correlation coefficient is denoted as r_{ab} between variable a and variable b, the first-order partial correlation coefficient between a and b conditioning on c is:

$$r_{ab\cdot c} = \frac{r_{ab} - r_{ac}r_{bc}}{\sqrt{(1 - r_{ac}^2)(1 - r_{bc}^2)}} \tag{2.22}$$

The second-order partial correlation coefficient between a and b conditioning on c and d is:

$$r_{ab\cdot cd} = \frac{r_{ab\cdot c} - r_{ad\cdot c}r_{bd\cdot c}}{\sqrt{(1 - r_{ad\cdot c}^2)(1 - r_{bd\cdot c}^2)}} \tag{2.23}$$

In pCastNet, three types of associations will be considered for a pair of genes: gene-gene (GG) association, exon-gene (EG) association, and exon-exon (EE) association. For GG association, the Pearson correlation coefficient is calculated between gene 1 (g_1) and gene 2 (g_2) and denoted as $r_{g_1 g_2}$. For EG association, considering an exon (e_1) of gene 1 and gene 2 (g_2), besides the Pearson correlation coefficient $r_{e_1 g_2}$, the first-order partial correlation coefficient between e_1 and g_2 conditioning on gene 1 (g_1) is also calculated as $r_{e_1 g_2 \cdot g_1}$. The partial correlation can be interpreted as the association between e_1 and g_2 after removing the effect of g_1. If the partial correlation is high, the association between e_1 and g_2 is not due to the correlation between g_1 and g_2. For EE association, the correlation between an exon (e_1) of gene 1 and an exon (e_2) of gene 2 is calculated as $r_{e_1 e_2}$. The partial correlations $r_{e_1 e_2 \cdot g_1}$, $r_{e_1 e_2 \cdot g_2}$, and the second-order partial correlation coefficient $r_{e_1 e_2 \cdot g_1 g_2}$ can also be calculated to exclude the possibility that the EE correlation is due to the EG or the GG correlation. In summary, if the p-value for $r_{g_1 g_2}$ is significant, a GG link between gene 1 and gene 2 can be declared. If the p-values for both $r_{e_1 g_2}$ and $r_{e_1 g_2 \cdot g_1}$ are significant, an EG link between e_1 and g_2 can be declared and the association is not due to GG association. If the p-values for $r_{e_1 e_2}$, $r_{e_1 e_2 \cdot g_1}$,$r_{e_1 e_2 \cdot g_2}$, and $r_{e_1 g_2 \cdot g_1 g_2}$ are significant, an EE link between the two exons e1 and e2 can be declared, and the association is not due to GG or EG associations.

The authors used the approach proposed by Efron [63] to control the expected FDR conditioning on a dependence effect parameter A. The sparseness of a network was estimated according to the conditional FDR and a threshold on the sparseness was then chosen. The sparseness of a network is defined as the percentage of true links among all possible node pairs. The threshold selection has several advantages: first, the corresponding correlation thresholds are data dependent; second, we can derive an accurate estimate of the number of falsely declared links taking into consideration the dependence among hypotheses; and third, we can integrate prior information about the sparseness of networks if this information is available.

By applying pCastNet to exon arrays in 11 human tissues, the authors found that gene pairs with exon-gene or exon-exon links tend to have similar functions or are present in the same pathways. More interestingly, gene pairs with exon-gene or exon-exon links tend to share cis-elements in promoter regions and microRNA binding elements in $3'$ untranslated regions, which suggests the coupling of co-alternative-splicing, co-transcription-factor-binding, and co-microRNA-binding.

2.5 Alternative Splicing, Genetic Variation, and Disease

Because of its important role in gene regulation, malfunction of alternative splicing has contributed to many human diseases [64–66]. Among point mutations associated with human genetic diseases in the Human Gene Mutation Database, about 9.5% of them are within splice sites and may cause RNA splicing defects [67]. In addition, many disease mutations that target synonymous and nonsynonymous amino acid codon positions often affect the exon splicing and cause function defects. It was estimated that as many as 50% of disease mutations in exons affect splicing [68]. Differential alternative splicing studies have been performed in many diseases such as cancers. For instances, altered transcript isoform levels have been detected for many genes in prostate and breast cancer without significant changes in total transcript abundance [69, 70]. In addition, a study of Hodgkin lymphoma tumors using custom alternative splicing microarrays found that the relative abundance of alternatively spliced isoforms correlates with transformation and tumor grade [71]. These studies suggest that alternative splicing profiling may provide additional tools for tumor diagnosis.

Kwan et al. also studied the heritability of alternative splicing in healthy people [72]. They investigated the alternative splicing variation among humans using exon array profiling in lymphoblastoid cell lines derived from the CEU HapMap population. Through family-based linkage studies and allelic association studies, they identified marker loci linked to particular alternative splicing events. They detected both annotated and novel alternatively spliced variants, and that such variation among individuals is heritable and genetically controlled.

2.6 Online Resources

At the end of this chapter, we provide a list of online databases for alternative splicing in Table 2.1. These databases collect alternative splicing events in different organisms or study the effect of alternative splicing on protein structures, RT-PCR, and so on.

2.7 Summary

Alternative splicing has been realized as one of the most important gene regulatory mechanisms. The related research has been reinvigorated by the availability of large amount of sequence data and high-throughput technologies. Nevertheless, many important questions regarding the function, the mechanism, and the regulation of alternative splicing remain unanswered. The statistical and computational analysis of alternative splicing has also emerged as an important and relatively new field.

Table 2.1 Online databases for alternative splicing

Database	Description	Link
ASTD [50]	human, mouse, and rat	http://www.ebi.ac.uk/astd/main.html
PALSdb [73]	human, mouse, and worm	http://ymbc.ym.edu.tw/palsdb/
SpliceInfo [74]	human	http://spliceinfo.mbc.nctu.edu.tw/
ASmodeler [75]	human, mouse, and rat	http://genome.ewha.ac.kr/ECgene/ASmodeler/
ECgene [76]	human, mouse, rat, dog, zebrafish, fruit fly, chick, rhesus, and C. elegans	http://genome.ewha.ac.kr/ECgene/
ASG [77]	human	http://statgen.ncsu.edu/asg/
DEDB [78]	fruit fly	http://proline.bic.nus.edu.sg/dedb/
EuSplice [79]	23 eukaryotes	http://66.170.16.154/EuSplice
ASPicDB [80]	human	http://t.caspur.it/ASPicDB/
HOLLYWOOD [81]	human and mouse	http://hollywood.mit.edu
AS-ALPS [82]	the effects of alternative splicing on protein structure, interaction and network in human and mouse	http://as-alps.nagahama-i-bio.ac.jp
SpliceCenter [83]	the impact of alternative splicing on RT-PCR, RNAi, microarray, and peptide-based studies	http://discover.nci.nih.gov/splicecenter
SpliVaP [84]	changes in signatures among protein isoforms due to alternative splicing	http://www.bioinformatica.crs4.org/tools/dbs/splivap/

They will provide valuable information about the precisely regulated alternative splicing process and help us to advance our knowledge about the post-transcriptional regulation.

References

1. Wang, E. T., Sandberg, R., Luo, S., Khrebtukova, I., Zhang, L., Mayr, C., Kingsmore, S. F., Schroth, G. P., & Burge, C. B. (2008). Alternative isoform regulation in human tissue transcriptomes. *Nature*, *456*, 470–476.
2. Pan, Q., Shai, O., Lee, L. J., Frey, B. J., & Blencowe, B. J. (2008). Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nature Genetics*, *40*, 1413–1415.
3. Gilbert, W. (1978). Why genes in pieces? *Nature*, *271*, 501.
4. Breitbart, R. E., Andreadis, A., & Nadal-Ginard, B. (1987). Alternative splicing: A ubiquitous mechanism for the generation of multiple protein isoforms from single genes. *Annual Review of Biochemistry*, *56*, 467–495.
5. Johnson, J. M., Castle, J., Garrett-Engele, P., Kan, Z., Loerch, P. M., Armour, C. D., Santos, R., Schadt, E. E., Stoughton, R., & Shoemaker, D. D. (2003). Genome-wide survey of human alternative pre-mRNA splicing with exon junction microarrays. *Science*, *302*, 2141–2144.

6. Kan, Z., Rouchka, E. C., Gish, W. R., & States, D. J. (2001). Gene structure prediction and alternative splicing analysis using genomically aligned ESTs. *Genome Research*, *11*, 889–900.
7. Mironov, A. A., Fickett, J. W., & Gelfand, M. S. (1999). Frequent alternative splicing of human genes. *Genome Research*, *9*, 1288–1293.
8. Modrek, B., Resch, A., Grasso, C., & Lee, C. (2001). Genome-wide detection of alternative splicing in expressed sequences of human genes. *Nucleic Acids Research*, *29*, 2850–2859.
9. Graveley, B. R., Kaur, A., Gunning, D., Zipursky, S. L., Rowen, L., & Clemens, J. C. (2004). The organization and evolution of the dipteran and hymenopteran Down syndrome cell adhesion molecule (Dscam) genes. *RNA*, *10*, 1499–1506.
10. Missler, M., & Sudhof, T. C. (1998). Neurexins: Three genes and 1001 products. *Trends in Genetics*, *14*, 20–26.
11. Zdobnov, E. M., von Mering, C., Letunic, I., Torrents, D., Suyama, M., Copley, R. R., Christophides, G. K., Thomasova, D., Holt, R. A., Subramanian, G. M., Mueller, H. M., Dimopoulos, G., Law, J. H., Wells, M. A., Birney, E., Charlab, R., Halpern, A. L., Kokoza, E., Kraft, C. L., Lai, Z., Lewis, S., Louis, C., Barillas-Mury, C., Nusskern, D., Rubin, G. M., Salzberg, S. L., Sutton, G. G., Topalis, P., Wides, R., Wincker, P., Yandell, M., Collins, F. H., Ribeiro, J., Gelbart, W. M., Kafatos, F. C., & Bork, P. (2002). Comparative genome and proteome analysis of Anopheles gambiae and Drosophila melanogaster. *Science*, *298*, 149–159.
12. Kent, W. J. (2002). BLAT – the BLAST-like alignment tool. *Genome Research*, *12*, 656–664.
13. Florea, L., Hartzell, G., Zhang, Z., Rubin, G. M., & Miller, W. (1998). A computer program for aligning a cDNA sequence with a genomic DNA sequence. *Genome Research*, *8*, 967–974.
14. Wu, T. D., & Watanabe, C. K. (2005). GMAP: A genomic mapping and alignment program for mRNA and EST sequences. *Bioinformatics*, *21*, 1859–1875.
15. van Nimwegen, E., Paul, N., Sheridan, R., & Zavolan, M. (2006). SPA: A probabilistic algorithm for spliced alignment. *PLoS Genetics*, *2*, e24.
16. Florea, L., Di Francesco, V., Miller, J., Turner, R., Yao, A., Harris, M., Walenz, B., Mobarry, C., Merkulov, G. V., Charlab, R., Dew, I., Deng, Z., Istrail, S., Li, P., & Sutton, G. (2005). Gene and alternative splicing annotation with AIR. *Genome Research*, *15*, 54–66.
17. Heber, S., Alekseyev, M., Sze, S. H., Tang, H., & Pevzner, P. A. (2002). Splicing graphs and EST assembly problem. *Bioinformatics*, *18*(Suppl 1), S181–S188.
18. Kim, N., Shin, S., & Lee, S. (2005). ECgene: Genome-based EST clustering and gene modeling for alternative splicing. *Genome Research*, *15*, 566–576.
19. Xing, Y., Yu, T., Wu, Y. N., Roy, M., Kim, J., & Lee, C. (2006). An expectation-maximization algorithm for probabilistic reconstructions of full-length isoforms from splice graphs. *Nucleic Acids Research*, *34*, 3150–3160.
20. Leparc, G. G., & Mitra, R. D. (2007). Non-EST-based prediction of novel alternatively spliced cassette exons with cell signaling function in Caenorhabditis elegans and human. *Nucleic Acids Research*, *35*, 3192–3202.
21. Sorek, R., & Ast, G. (2003). Intronic sequences flanking alternatively spliced exons are conserved between human and mouse. *Genome Research*, *13*, 1631–1637.
22. Sorek, R., Shemesh, R., Cohen, Y., Basechess, O., Ast, G., & Shamir, R. (2004). A non-EST-based method for exon-skipping prediction. *Genome Research*, *14*, 1617–1623.
23. Yeo, G. W., Van Nostrand, E., Holste, D., Poggio, T., & Burge, C. B. (2005). Identification and analysis of alternative splicing events conserved in human and mouse. *Proceedings of the National Academy of Sciences of the United States of America*, *102*, 2850–2855.
24. Chen, L., & Zheng, S. (2008). Identify alternative splicing events based on position-specific evolutionary conservation. *PLoS One*, *3*, e2806.
25. Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32.
26. Parmley, J. L., Urrutia, A. O., Potrzebowski, L., Kaessmann, H., & Hurst, L. D. (2007). Splicing and the evolution of proteins in mammals. *PLoS Biology*, *5*, e14.
27. Fairbrother, W. G., Holste, D., Burge, C. B., & Sharp, P. A. (2004). Single nucleotide polymorphism-based validation of exonic splicing enhancers. *PLoS Biology*, *2*, e268.
28. Boutz, P. L., Stoilov, P., Li, Q., Lin, C. H., Chawla, G., Ostrow, K., Shiue, L., Ares, M., Jr., & Black, D. L. (2007). A post-transcriptional regulatory switch in polypyrimidine tract-binding proteins reprograms alternative splicing in developing neurons. *Genes & Development*, *21*, 1636–1652.

29. Clark, T. A., Schweitzer, A. C., Chen, T. X., Staples, M. K., Lu, G., Wang, H., Williams, A., & Blume, J. E. (2007). Discovery of tissue-specific exons using comprehensive human exon microarrays. *Genome Biology*, *8*, R64.
30. Yeo, G. W., Xu, X., Liang, T. Y., Muotri, A. R., Carson, C. T., Coufal, N. G., & Gage, F. H. (2007). Alternative splicing events identified in human embryonic stem cells and neural progenitors. *PLoS Computational Biology*, *3*, 1951–1967.
31. Castle, J. C., Zhang, C., Shah, J. K., Kulkarni, A. V., Kalsotra, A., Cooper, T. A., & Johnson, J. M. (2008). Expression of 24,426 human alternative splicing events and predicted cis regulation in 48 tissues and cell lines. *Nature Genetics*, *40*, 1416–1425.
32. Clark, T. A., Sugnet, C. W., & Ares, M., Jr. (2002). Genomewide analysis of mRNA processing in yeast using splicing-specific microarrays. *Science*, *296*, 907–910.
33. Cline, M. S., Blume, J., Cawley, S., Clark, T. A., Hu, J. S., Lu, G., Salomonis, N., Wang, H., & Williams, A. (2005). ANOSVA: A statistical method for detecting splice variation from expression data. *Bioinformatics*, *21*(Suppl. 1), i107–i115.
34. Purdom, E., Simpson, K. M., Robinson, M. D., Conboy, J. G., Lapuk, A. V., & Speed, T. P. (2008). FIRMA: A method for detection of alternative splicing from exon array data. *Bioinformatics*, *24*, 1707–1714.
35. Wang, H., Hubbell, E., Hu, J. S., Mei, G., Cline, M., Lu, G., Clark, T., Siani-Rose, M. A., Ares, M., Kulp, D. C., & Haussler, D. (2003). Gene structure-based splice variant deconvolution using a microarray platform. *Bioinformatics*, *19*(Suppl. 1), i315–i322.
36. Li, C., & Wong, W. H. (2001). Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proceedings of the National Academy of Sciences of the United States of America*, *98*, 31–36.
37. Anton, M. A., Gorostiaga, D., Guruceaga, E., Segura, V., Carmona-Saez, P., Pascual-Montano, A., Pio, R., Montuenga, L. M., & Rubio, A. (2008). SPACE: An algorithm to predict and quantify alternatively spliced isoforms using microarrays. *Genome Biology*, *9*, R46.
38. Shai, O., Morris, Q. D., Blencowe, B. J., & Frey, B. J. (2006). Inferring global levels of alternative splicing isoforms using a generative model of microarray data. *Bioinformatics*, *22*, 606–613.
39. Pan, Q., Shai, O., Misquitta, C., Zhang, W., Saltzman, A. L., Mohammad, N., Babak, T., Siu, H., Hughes, T. R., Morris, Q. D., Frey, B. J., & Blencowe, B. J. (2004). Revealing global regulatory features of mammalian alternative splicing using a quantitative microarray platform. *Molecular Cell*, *16*, 929–941.
40. Fagnani, M., Barash, Y., Ip, J. Y., Misquitta, C., Pan, Q., Saltzman, A. L., Shai, O., Lee, L., Rozenhek, A., Mohammad, N., Willaime-Morawek, S., Babak, T., Zhang, W., Hughes, T. R., van der Kooy, D., Frey, B. J., & Blencowe, B. J. (2007). Functional coordination of alternative splicing in the mammalian central nervous system. *Genome Biology*, *8*, R108.
41. Nagalakshmi, U., Wang, Z., Waern, K., Shou, C., Raha, D., Gerstein, M., & Snyder, M. (2008). The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science*, *320*, 1344–1349.
42. Wilhelm, B. T., Marguerat, S., Watt, S., Schubert, F., Wood, V., Goodhead, I., Penkett, C. J., Rogers, J., & Bahler, J. (2008). Dynamic repertoire of a eukaryotic transcriptome surveyed at single-nucleotide resolution. *Nature*, *453*, 1239–1243.
43. Lister, R., O'Malley, R. C., Tonti-Filippini, J., Gregory, B. D., Berry, C. C., Millar, A. H., & Ecker, J. R. (2008). Highly integrated single-base resolution maps of the epigenome in Arabidopsis. *Cell*, *133*, 523–536.
44. Cloonan, N., Forrest, A. R., Kolle, G., Gardiner, B. B., Faulkner, G. J., Brown, M. K., Taylor, D. F., Steptoe, A. L., Wani, S., Bethel, G., Robertson, A. J., Perkins, A. C., Bruce, S. J., Lee, C. C., Ranade, S. S., Peckham, H. E., Manning, J. M., McKernan, K. J., & Grimmond, S. M. (2008). Stem cell transcriptome profiling via massive-scale mRNA sequencing. *Nature Methods*, *5*, 613–619.
45. Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., & Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*, *5*, 621–628.

46. Marioni, J. C., Mason, C. E., Mane, S. M., Stephens, M., & Gilad, Y. (2008). RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, *18*, 1509–1517.
47. Sultan, M., Schulz, M. H., Richard, H., Magen, A., Klingenhoff, A., Scherf, M., Seifert, M., Borodina, T., Soldatov, A., Parkhomchuk, D., Schmidt, D., O'Keeffe, S., Haas, S., Vingron, M., Lehrach, H., & Yaspo, M. L. (2008). A global view of gene activity and alternative splicing by deep sequencing of the human transcriptome. *Science*, *321*, 956–960.
48. Jiang, H., & Wong, W. H. (2009). Statistical inferences for isoform expression in RNA-Seq. *Bioinformatics*,*25*, 1026–1032.
49. Zheng, S., & Chen, L. (2009). A hierarchical Bayesian model for comparing transcriptomes at the individual transcript isoform level. *Nucleic Acids Research*, *37*,e75.
50. Stamm, S., Riethoven, J. J., Le Texier, V., Gopalakrishnan, C., Kumanduri, V., Tang, Y., Barbosa-Morais, N. L., & Thanaraj, T. A. (2006). ASD: A bioinformatics resource on alternative splicing. *Nucleic Acids Research*, *34*, D46–D55.
51. Zhou, Z., Licklider, L. J., Gygi, S. P., & Reed, R. (2002). Comprehensive proteomic analysis of the human spliceosome. *Nature*, *419*, 182–185.
52. Jurica, M. S., & Moore, M. J. (2003). Pre-mRNA splicing: Awash in a sea of proteins. *Molecular Cell*, *12*, 5–14.
53. Wang, Z., & Burge, C. B. (2008). Splicing regulation: From a parts list of regulatory elements to an integrated splicing code. *RNA*, *14*, 802–813.
54. Huh, G. S., & Hynes, R. O. (1994). Regulation of alternative pre-mRNA splicing by a novel repeated hexanucleotide element. *Genes & Development*, *8*, 1561–1574.
55. McCullough, A. J., & Berget, S. M. (1997). G triplets located throughout a class of small vertebrate introns enforce intron borders and regulate splice site selection. *Molecular Cell Biology*, *17*, 4562–4571.
56. Chou, M. Y., Underwood, J. G., Nikolic, J., Luu, M. H., & Black, D. L. (2000). Multisite RNA binding and release of polypyrimidine tract binding protein during the regulation of c-src neural-specific splicing. *Molecular Cell*, *5*, 949–957.
57. Wang, Z., Rolish, M. E., Yeo, G., Tung, V., Mawson, M., & Burge, C. B. (2004). Systematic identification and analysis of exonic splicing silencers. *Cell*, *119*, 831–845.
58. Zhang, X. H., & Chasin, L. A. (2004). Computational definition of sequence motifs governing constitutive exon splicing. *Genes & Development*, *18*, 1241–1250.
59. Black, D. L. (2003). Mechanisms of alternative pre-messenger RNA splicing. *Annual Review of Biochemistry*, *72*, 291–336.
60. Matlin, A. J., Clark, F., & Smith, C. W. (2005). Understanding alternative splicing: Towards a cellular code. *Nature Review. Molecular Cell Biology*, *6*, 386–398.
61. Fu, X. D. (2004). Towards a splicing code. *Cell*, *119*, 736–738.
62. Chen, L., & Zheng, S. (2009). Studying alternative splicing regulatory networks through partial correlation analysis. *Genome Biology*, *10*, R3.
63. Efron, B. (2007). Correlation and large-scale simultaneous significance testing. *Journal of the American Statistical Association*, *102*, 93–103.
64. Faustino, N. A., & Cooper, T. A. (2003). Pre-mRNA splicing and human disease. *Genes & Development*, *17*, 419–437.
65. Garcia-Blanco, M. A., Baraniak, A. P., & Lasda, E. L. (2004). Alternative splicing in disease and therapy. *Nature Biotechnology*, *22*, 535–546.
66. Blencowe, B. J. (2000). Exonic splicing enhancers: Mechanism of action, diversity and role in human genetic diseases. Trends in Biochemical Sciences, *25*, 106–110.
67. Krawczak, M., Thomas, N. S., Hundrieser, B., Mort, M., Wittig, M., Hampe, J., & Cooper, D. N. (2007). Single base-pair substitutions in exon-intron junctions of human genes: Nature, distribution, and consequences for mRNA splicing. *Human Mutation*, *28*, 150–158.
68. Blencowe, B. J. (2006). Alternative splicing: New insights from global analyses. *Cell*, *126*, 37–47.
69. Li, H. R., Wang-Rodriguez, J., Nair, T. M., Yeakley, J. M., Kwon, Y. S., Bibikova, M., Zheng, C., Zhou, L., Zhang, K., Downs, T., Fu, X. D., & Fan, J. B. (2006). Two-dimensional transcriptome profiling: Identification of messenger RNA isoform signatures in prostate cancer from archived paraffin-embedded cancer specimens. *Cancer Research*, *66*, 4079–4088.

70. Li, C., Kato, M., Shiue, L., Shively, J. E., Ares, M., Jr., & Lin, R. J. Cell type and culture condition-dependent alternative splicing in human breast cancer cells revealed by splicing-sensitive microarrays. *Cancer Research*, *66*, 1990–1999 (2006).
71. Relogio, A., Ben-Dov, C., Baum, M., Ruggiu, M., Gemund, C., Benes, V., Darnell, R. B., & Valcarcel, J. (2005). Alternative splicing microarrays reveal functional expression of neuron-specific regulators in Hodgkin lymphoma cells. *The Journal of Biological Chemistry*, *280*, 4779–4784.
72. Kwan, T., Benovoy, D., Dias, C., Gurd, S., Serre, D., Zuzan, H., Clark, T. A., Schweitzer, A., Staples, M. K., Wang, H., Blume, J. E., Hudson, T. J., Sladek, R., & Majewski, J. (2007). Heritability of alternative splicing in the human genome. *Genome Research*, *17*, 1210–1218.
73. Huang, Y. H., Chen, Y. T., Lai, J. J., Yang, S. T., & Yang, U. C. (2002). PALS db: Putative Alternative Splicing database. *Nucleic Acids Research*, *30*, 186–190.
74. Huang, H. D., Horng, J. T., Lin, F. M., Chang, Y. C., & Huang, C. C. (2005). SpliceInfo: An information repository for mRNA alternative splicing in human genome. *Nucleic Acids Research*, *33*, D80–D85.
75. Kim, N., Shin, S., & Lee, S. (2004). ASmodeler: Gene modeling of alternative splicing from genomic alignment of mRNA, EST and protein sequences. *Nucleic Acids Research*, *32*, W181–W186.
76. Kim, P., Kim, N., Lee, Y., Kim, B., Shin, Y., & Lee, S. (2005). ECgene: Genome annotation for alternative splicing. *Nucleic Acids Research*, *33*, D75–D79.
77. Leipzig, J., Pevzner, P., & Heber, S. (2004). The Alternative Splicing Gallery (ASG): Bridging the gap between genome and transcriptome. *Nucleic Acids Research*, *32*, 3977–3983.
78. Lee, B. T., Tan, T. W., & Ranganathan, S. (2004). DEDB: A database of Drosophila melanogaster exons in splicing graph form. *BMC Bioinformatics*, *5*, 189.
79. Bhasi, A., Pandey, R. V., Utharasamy, S. P., & Senapathy, P. (2007). EuSplice: A unified resource for the analysis of splice signals and alternative splicing in eukaryotic genes. *Bioinformatics*, *23*, 1815–1823.
80. Castrignano, T., D'Antonio, M., Anselmo, A., Carrabino, D., D'Onorio De Meo, A., D'Erchia, A. M., Licciulli, F., Mangiulli, M., Mignone, F., Pavesi, G., Picardi, E., Riva, A., Rizzi, R., Bonizzoni, P., & Pesole, G. (2008). ASPicDB: A database resource for alternative splicing analysis. *Bioinformatics*, *24*, 1300–1304.
81. Holste, D., Huo, G., Tung, V., & Burge, C. B. (2006). HOLLYWOOD: A comparative relational database of alternative splicing. *Nucleic Acids Research*, *34*, D56–D62.
82. Shionyu, M., Yamaguchi, A., Shinoda, K., Takahashi, K., & Go, M. (2009). AS-ALPS: A database for analyzing the effects of alternative splicing on protein structure, interaction and network in human and mouse. *Nucleic Acids Research*, *37*, D305–D309.
83. Ryan, M. C., Zeeberg, B. R., Caplen, N. J., Cleland, J. A., Kahn, A. B., Liu, H., & Weinstein, J. N. (2008). SpliceCenter: A suite of web-based bioinformatic applications for evaluating the impact of alternative splicing on RT-PCR, RNAi, microarray, and peptide-based studies. *BMC Bioinformatics*, *9*, 313.
84. Floris, M., Orsini, M., & Thanaraj, T. A. (2008). Splice-mediated Variants of Proteins (SpliVaP) – data and characterization of changes in signatures among protein isoforms due to alternative splicing. *BMC Genomics*, *9*, 453.

Chapter 3
Statistical Learning and Modeling of TF-DNA Binding

Bo Jiang and Jun S. Liu

Abstract Discovering binding sites and motifs of specific TFs is an important first step towards the understanding of gene regulation circuitry. Computational approaches have been developed to identify transcription factor binding sites from a set of co-regulated genes. Recently, the abundance of gene expression data, ChIP-based TF-binding data (ChIP-array/seq), and high-resolution epigenetic maps have brought up the possibility of capturing sequence features relevant to TF-DNA interactions so as to improve the predictive power of gene regulation modeling. In this chapter, we introduce some statistical models and computational strategies used to predict TF-DNA interactions from the DNA sequence information, and describe a general framework of predictive modeling approaches to the TF-DNA binding problem, which includes both traditional regression methods and statistical learning methods by selecting relevant sequence features and epigenetic markers.

3.1 Introduction

The linear biopolymers, DNA, RNA, and proteins, are the three central molecular building blocks of life. DNA is an information storage molecule. All of the hereditary information of an individual organism is contained in its genome, which consists of sequences of the four DNA bases (nucleotides), A, C, G, and T. RNA has a wide variety of roles, including a small but important set of functions. Proteins, which are chains of 20 different amino acid residues, are the action molecules of life, being responsible for nearly all the functions of all living beings and forming many of life's structures. All protein sequences are coded by segments of the genome called genes. How genetic information flows from DNA to RNA and then to

B. Jiang (✉)
Department of Statistic, Harvard University, Cambridge, MA 02138, USA
e-mail: bjiang@fas.harvard.edu

J.S. Liu
Department of Statistic, Harvard University, Cambridge, MA 02138, USA
e-mail: jliu@stat.harvard.edu

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_3,
© Springer-Verlag Berlin Heidelberg 2011

protein is regarded as the central dogma of molecular biology. Genome sequencing projects with emergence of microarray techniques have resulted in rapidly growing and publicly available databases of DNA and protein sequences, structures, and genome-wide expression. One of the most interesting questions scientists try to answer is to understand the mechanism of transcribing DNA sequence information into messenger RNA (mRNA), which is used as templates to produce protein molecules, and other types of RNA molecules. This process is generally known as the transcriptional regulation of genes.

A substantial portion of a cell's morphological and functional attributes is determined at the level of gene transcription. In eukaryotes, transcription is initiated by the binding of RNA polymerase II to the core promoters of genes, which reads the sequence of one strand of the DNA and synthesizes mRNA. The efficiency of transcription is regulated by proteins called transcription factors (TFs) binding to their recognition sites located mostly upstream of the genes (promoter regions), but also not infrequently downstream or intronic regions. Transcription factor binding sites (TFBSs) are short sequence segments ($\sim$8–20 base pairs long) located near genes' transcription start sites (TSSs). TFBSs usually show a conserved pattern, which is often called a TF binding motif (TFBM). Discovering binding sites and motifs of specific TFs is an important first step towards the understanding of gene regulation circuitry. In the past two decades, computational approaches have been developed to identify TFBSs from a set of genes that are possibly co-regulated or are mutual orthologs from different species. The gene co-regulation information can be obtained from analyzing mRNA expression microarrays, and orthologous genes can be detected via comparative genomics approaches. Recently, the abundance of gene expression data, ChIP-based TF-binding data (ChIP-array/seq), and high-resolution epigenetic maps have brought up the possibility of capturing sequence features relevant to TF-DNA interactions so as to improve the predictive power of gene regulation modeling.

In this chapter, we describe some statistical methods and models used to predict TF-DNA interactions from the DNA sequence information. Section 3.2 gives an overview of experimental methods for identifying TF binding sites. Section 3.3 introduces statistical models and computational strategies for finding TF binding motifs from experimental data. Section 3.4 describes a general framework of predictive modeling approaches to the TF-DNA binding problem, which includes both traditional regression methods and statistical learning methods by selecting relevant sequence features and epigenetic markers. Section 3.5 concludes the chapter with a brief discussion.

3.2 Experimental Methods for Identifying TF-DNA Interactions

The initial determination of the binding site of a specific TF to a DNA site is achieved using laboratory assays such as electrophoretic mobility shift assays and DNase footprinting. However, these experiments can only locate TFBSs on a gene-by-gene and site-by-site basis, and are laborious, time-consuming, and unsuitable for large scale studies.

With the availability of complete genome sequences, biologists can now use techniques such as DNA gene expression microarrays to measure the expression level of each gene in an organism under various conditions. A collection of expressions of each gene measured under various conditions is called the gene expression profile. Genes can be divided into clusters according to similarities in their expression profiles-genes in the same cluster respond similarly to environmental and developmental changes and thus may be co-regulated by the same TF or the same group of TFs. On the other hand, the complete genome sequences of many organisms permit comprehensive comparative analysis of genome structures. By utilizing the fact that the genes that code for the same protein in related species are likely to be similarly regulated, cross-species comparison provides another means to identify multiple genes that are likely to be regulated similarly. Computational approaches have been developed to search for TFBSs in the upstream of genes in particular clusters revealed by microarray expression analyses or comparative genomic analyses.

Recent years have seen rapid innovations of TF binding assays such as ChIP-array/seq and protein binding microarray technology. Chromatin immunoprecipitation followed by microarray (ChIP-array) or massively parallel DNA sequencing (ChIP-seq) technology can measure where a particular TF binds to DNA in the whole genome under a given experimental condition at a resolution from a few hundred bases (ChIP-array) to few tens of bases (ChIP-seq). Although computational analysis is still required to pinpoint the short binding sites of the transcription factor from all potential TF binding regions, the newly developed technologies can help quantify the specificity with which TFs recognize their DNA target sites. To allow for high-throughput characterization of the DNA binding site sequence specificities of TFs in a rapid and universal manner, a novel DNA microarray-based in vitro technology, termed protein binding microarrays (PBMs), has been developed [2]. In PBM assay, epitope-tagged TFs were bound directly to double strand DNA spotted on a compact, universal microarray that contains all possible sequence variants of a given length. PBM assays permit the discovery of subtle preferences in transcription factor binding sites (including interdependencies among different positions) and can be used with transcription factors from any species regardless of the level to which its genome has been characterized. However, PBM also has limitations itself. For example, some transcription factors have to be modified after translation process in order to potentially interact with DNA and PBM cannot capture such interactions as an *in vitro* assay.

Although transcription factors are known to have a high affinity to specific DNA sequences, DNA sequence alone is a poor predictor of genome wide transcription factor binding locations. The structure of chromatin is likely to play a significant role in defining the precise genomic locations that are accessible to transcription factors for binding. Detailed nucleosome occupancy maps have been generated by ChIP-array [27] or by ChIP-seq experiments [16]. The characterization of nucleosome occupancy dynamics with DNA sequences and the hierarchical organization of chromatin enable us to infer cell-type and condition dependent TF-DNA interactions and their regulatory roles.

3.3 Generative Models for Discovering TF Binding Motifs

The goal of statistical motif finding is to look for common sequence segments associated with regulatory or binding response measurements such as gene expression values or ChIP-array/seq fold changes. In this section, we will introduce a few generative models of binding motifs, and the block-motif model, a statistical model that can be handled efficiently by Gibbs sampling to discover TF binding motifs enriched in a set of sequences. The basic framework can be further extended to integrate comparative genomic information, and to jointly model the binding of multiple TFs to account for their synergistic interactions. Finally, we describe a computational strategy to discover binding motifs from ChIP-array technology by optimizing a scoring function derived from a generative model.

3.3.1 Motif Formulations and General Discovery Strategies

There are two ways of discovering novel binding sites of a TF: scanning methods and de novo methods. In a scanning method, one uses a motif representation resulting from experimentally determined binding sites to scan the genome sequence to find more matches. In de novo methods, one attempts to find novel motifs that are "enriched" in a set of sequences. This section focuses on the latter class of methods. The de novo methods can also be divided into two classes, according roughly to two general data formulations for representing a motif: the consensus sequence or a position-specific weight matrix (PSWM).

The simplest method for finding a motif is to check for the over-representation of every oligonucleotide of a given length (i.e., every k-mer). However, binding sites of a TF are usually "very badly spelled", and can tolerate many "typos". Hence, degenerated IUPAC symbols for ambiguous bases are frequently used in the consensus analysis. Unlike consensus analysis, which only reflects most frequently occurred base types at each motif position without an explicit account of frequencies, statistical models based on a probabilistic representation of the preference of nucleotides at each position is generally more informative. In principle, TF can and does bind to any DNA sequence with a probability dependent on the binding energy. Thus, it is desirable to model TF/DNA interaction based on statistical mechanics, which provides a theoretical support for the use of a probabilistic representation. The most widely used statistical model of motif is the PSWM, or a product multinomial distribution [19]. The columns of a PSWM describe the occurrence frequencies of the four nucleotides in the corresponding motif position. Figure 3.1 shows a convenient graphical representation of the PSWM: the sequence logo plot, in which the height

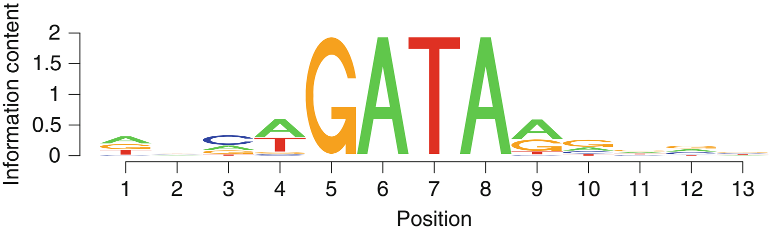


Fig. 3.1 A sequence logo plot from the PSWM of a GATA binding factor

of each column is proportional to its information content and the size of each letter is proportional to its frequency at that position.

In general, there are two types of approaches for statistical binding motif discovery based on different representations of motifs: those proceed by enumerating regular expressions, such as YMF [23] and those proceed by iteratively updating PSWM, such as Consensus [11], MEME [1], and Gibbs motif sampler [15, 18].

3.3.2 A Block-Motif Model for Finding Motifs in a Set of Sequences

Our focus here is the discovery of binding motifs in a given set of DNA sequences. The main motivation is that repetitive patterns in the upstream regions of co-regulated genes may correspond to functional sites to which certain TF binds so as to control gene expressions.

The problem of motif finding can be formulated as a Bayesian missing data problem under the block-motif model [19] as shown in Fig. 3.1. The model says that at "missing" locations, $\mathbf{A} = (a_1, a_2, \ldots, a_k)$, there are repeated occurrences of a motif. So the sequence segments at these locations should look similar to each other. In other parts of the sequence, the residues (or base pairs) are modeled as independent and identically distributed observations from a multinomial distribution (these background residues can also be modeled as a kth order Markov chain). The background frequency $\boldsymbol{\theta_0} = (\theta_{0,A}, \theta_{0,C}, \theta_{0,G}, \theta_{0,T})$ can be assumed known, or estimated from the data in advance because the motif site positions are only a very tiny fraction of all the sequence positions. For a motif of width w, we let $\Theta = (\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \ldots, \boldsymbol{\theta_w})$, where each describes the base frequency at position j of the motif, and the matrix Θ is the PSWM for the motif. For a particular sequence s of length n, given motif location a, the likelihood can be written as:

$$\begin{aligned} p(s|\Theta, \boldsymbol{\theta_0}, a) &= p(s_{[1:(a-1)]}|\boldsymbol{\theta_0}) \times p(s_{[a:(a+w-1)]}|\Theta) \times p(s_{[(a+w):n]}|\boldsymbol{\theta_0}) \\ &= p(s_{[1:n]}|\boldsymbol{\theta_0}) \times \frac{p(s_{[a:(a+w-1)]}|\Theta)}{p(s_{[a:(a+w-1)]}|\boldsymbol{\theta_0})}. \end{aligned}$$

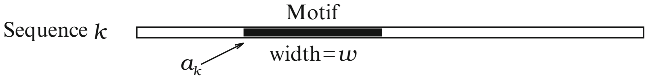


Fig. 3.2 A schematic plot of the block-motif model

Suppose we are given a set of sequences $\mathbf{S} = (s_1, s_2, \ldots, s_k)$, which share a common repeated motif. By integrating out Θ after incorporating the prior, we can get the likelihood for alignment vector:

$$p(\mathbf{S}|\mathbf{A}) = \int \left\{ \prod_{i=1}^{k} p(s_i|\mathbf{A}, \Theta, \boldsymbol{\theta}_0) \right\} d\Theta.$$

By assuming a uniform prior on $\mathbf{A}$, two different Gibbs sampling algorithms can be used to sample from the posterior distribution of $\mathbf{A}$: iterative sampling and predictive updating. The procedure of iterative sampling proceeds by iterating between sampling $\Theta^{(t)}$ from $p(\Theta|\mathbf{A}, \mathbf{S})$ and drawing $\mathbf{A}^{(t)}$ from $p(\mathbf{A}|\Theta^{(t)}, \mathbf{S})$. The predictive updating (PU) procedure is based on a "collapsed Gibbs sampler" that integrates out Θ and iteratively updates the a_j. More precisely, one can pretend that binding sites in all but the jth sequence have been found, and predict the binding site's location in the jth sequence by drawing $a_j^{(t)}$ from the predictive distribution:

$$p(a_j|\mathbf{A}_j^{(t-1)}, \mathbf{S}) \propto \prod_{i=1}^{w} \frac{q_{i,s_{j,a_j+i-1}}}{q_{0,s_{j,a_j+i-1}}},$$

where $q_{i,x} = (c_{i,x} + b_x)/(\sum_x (c_{i,x} + b_x))$, $c_{i,x}$ is the count of nucleotide type x at position i, $c_{0,x}$ is the count of nucleotide type x in all non-site positions, and b_x is the "pseudo-count" for nucleotide x.

The above model is not satisfactory in several aspects. First, some sequences often have multiple motif sites and some other sequences do not have any site at all. Thus, it is more reasonable to view the dataset as a long sequence that houses an unknown number of TF binding sites. The above model can be modified by adding an indicator (possible taking more than two values corresponding to multiple TFs) to each position of the sequences, and following the same predictive update strategy. Second, the distribution of the known TF binding site locations (e.g., distances between binding sites and the translation start site) can be formulated as an informed alignment prior. Third, non-coding sequences are often heterogeneous in compositions, in which case one needs to use higher-order Markov model or incorporate position-specific background model. In addition, the width w of a motif was assumed to be known and fixed; we may instead view w as an additional model parameter (in this case jointly sampling from posterior distribution is not easy since the dimensionality of Θ changes with w).

Further modeling efforts can also be made to incorporate biological considerations. The assumption in the product multinomial model is that all columns of a weight matrix are independent. Zhou and Liu [30] extended the independent weight matrix model to include one or more correlated column pairs. A Metropolis-Hastings step was added to the original Gibbs sampling algorithm so as to delete or add a pair of correlated columns periodically.

The main difficulties with motif finding in higher eukaryotes include the increased volume of the sequence search space, with proximal TFBSs a few kilobases away from the TSSs; the increased occurrence of low-complexity repeats; the increased complexity in regulatory controls due to TF-TF interactions; and shorter and less-conserved TFBSs. Despite these challenges, there are two possible redeeming factors: (i) many eukaryotic genomes have been or are being sequenced, and comparative genomic analysis can be extremely powerful; and (ii) most eukaryotic genes are controlled by a combination of a few factors with the corresponding binding sites forming homotypic or heterotypic clusters known as “cis-regulatory modules”.

3.3.3 Comparative Genomic Approach for TF Binding Sites Discovery

Transcription factor binding sites across species are more conserved than random background due to functional constraints. With the advent of whole genome sequencing, computational phylogenetic footprinting methods, involving cross-species comparison of DNA sequences, have emerged. Traditional “horizontal” approach requires a set of co-regulated genes to identify common motifs. In contrast, phylogenetic footprinting is a “vertical” approach, which uses orthologous genes in related species to identify common motifs for this gene across multiple species.

McCue et al. [21] introduced a phylogenetic footprinting method with application in proteobacterial genomes. The method begins with *E.coli* annotated gene, and applies tBlastn with stringent criteria to identify orthologous genes in eight other gamma proteobacterial species. Upstream intergenic regions from nine species are extracted, and a Gibbs motif sampler with the following important extensions is utilized. First, a motif model that accounts for palindromic patterns in TF-binding sites is employed. Because DNA sequences tend to have varying composition, a position-specific background model, estimated with a Bayesian segmentation algorithm, is used to contrast with the motif PSWM. Furthermore, the empirical distribution of spacing between TF-binding sites and the translation start site, observed from the *E.coli* genome sequence, is incorporated to improve the algorithm’s focus on more probable locations of binding sites. Lastly, the algorithm is configured to detect 0, 1 or 2 sites in each upstream region in a data set. The algorithm is applied to a study set of 184 *E.coli* genes whose promoters contain documented binding sites for 53 different TFs. Among the 184 most probable motif predictions, 146 corresponds to known binding sites. The remaining data sets contain several predictions with larger *maximum a posteriori probability* (MAP) values than the true sites, suggesting the possibility of undocumented regulatory sites in these data.

3.3.4 Hidden Markov Models for Cis-regulatory Module Discovery

Transcription regulation is controlled by coordinated binding of one or more transcription factors in the promoter regions of genes. In many species, especially higher eukaryotes, transcription factor binding sites tend to occur as homotypic or heterotypic clusters, also known as cis-regulatory modules (CRMs). The number of sites and distances between the sites, however, vary greatly in a module. One approach to locating CRMs is by predicting novel motifs and looking for co-occurrences [23]. However, since individual motifs in the cluster may not be well-conserved, such an approach often leads to a large number of false negatives. To cope with these difficulties, one can use Hidden Markov Models (HMMs) to capture both the spatial and contextual dependencies of the motifs in a CRM and use MCMC sampling to infer the CRM models and locations [24, 32]. Gupta and Liu [10] introduced a competing strategy, which first uses existing de novo motif finding algorithms and/or transcription factor (TF) databases to compose a list of putative binding motifs, $\Delta = \{\Theta_1, \ldots, \Theta_D\}$, where D is in the range of 50–100, and then simultaneously update these motifs and estimate the posterior probability for each of them to be included in the CRM.

Let $\mathbf{S}$ denote the set of n sequences with lengths, $L_1, L_2, \ldots, L_n$, respectively, corresponding to the upstream regions of n co-regulated genes. Assume that the CRM consists of K $(<D)$ different kinds of TFs with distinctive PSWMs. Both the PSWMs and K are unknown and need be inferred from the data. Let $\mathbf{a} = \{a_{ij}; i = 1, \ldots, n; j = 1, \ldots, L_i\}$, where a_{ij} denotes the location of the jth motif site (irrespective of motif type) in the ith sequence.

Associated with each site is its type indicator $T_{i,j}$, with $T_{i,j}$ taking one of the K values and let $\mathbf{T} = (T_{i,j})$. The dependence between $T_{i,j}$ and $T_{i,j+1}$ is modeled by a $K \times K$ transition matrix $\boldsymbol{\tau}$. The distance between neighboring TF binding sites in a CRM, $d_{i,j} = a_{i,j+1} - a_{i,j}$, is assumed to follow the distribution $Q(d; \lambda, w) = (1-\lambda)^{d-w+1}\lambda$ $(d = w, w+1, \ldots)$. The background sequence follows a multinomial distribution with unknown parameter $\boldsymbol{\rho} = (\rho_A, \rho_C, \rho_G, \rho_T)$. Finally, let $\mathbf{u}$ be a binary vector indicating which motifs are included in the module, i.e., $\mathbf{u} = (u_1, \ldots, u_D)^T$, where $u_j = 1$ if the jth motif type is present in the module, and 0 otherwise. By construction, $|\mathbf{u}| = K$. The set of PSWMs for the CRM is then $\Theta = \{\Theta_j : u_j = 1\}$.

Since now we restrict our inference of CRM to a subset of Δ, the probability model for the observed sequence data can be written out explicitly as in [10]. To implement the Bayesian analysis, we can prescribe a joint prior distribution on the unknown parameters, $\Omega = \{\Delta, \boldsymbol{\tau}, \lambda, \boldsymbol{\rho}\}$, and a prior probability of π for each $u_j = 1$. A Gibbs sampling approach was developed in [24] to sample both Ω and $\mathbf{u}$

from their joint posterior distribution. But given the flexibility of the model and the size of the parameter space for an unknown **u**, it is unlikely that a standard MCMC approach can converge to a good solution in a reasonable amount of time. If we ignore the ordering of sites **T** and assume components of **a** to be independent, this model is reduced to the original motif model, which can be updated through the previous Gibbs sampling procedure.

The following strategy was developed in [10]. With a starting set of putative binding motifs Δ, one iterates the following Monte Carlo sampling steps: (i) Given the current collection Δ of motif PSWMs (or sites), sample motifs into the CRM; (ii) Given the CRM configuration and the PSWMs, update the motif site locations through Gibbs sampling; and (iii) Given motif site locations, update the corresponding PSWMs and other parameters. Since the construction of a CRM in this formulation is by using an indicator variable Δ, it is natural to use a genetic-type algorithm to speed up computation. So an evolutionary Monte Carlo [17] strategy was implemented for the module inference, and very good results were obtained for a range of examples.

3.3.5 Motif Discovery in ChIP-Array Experiments

Chromatin immunoprecipitation followed by mRNA microarray analysis (ChIP-array) has become a popular procedure for studying genome-wide protein-DNA interactions and transcription regulation. However, it can only map the probable protein-DNA interaction loci within 1–2 kilo-bases resolution. Liu et al. [20] introduced a computational method, Motif Discovery scan (MDscan), that examines the ChIP-array selected sequences and searches for DNA sequence motifs representing the protein-DNA interaction sites. MDscan combines the advantages of two widely adopted motif search strategies, word enumeration and position-specific weight matrix updating, and incorporates the ChIP-array ranking information to accelerate searches and enhance their success rates.

Consider a set of n DNA sequences selected from ChIP-array experiments, ranked according to their ChIP-array enhancement scores, from the highest to the lowest. MDscan first scrutinizes the top t (e.g., 5–50) sequences in the ranking to form a set of promising candidates. Assuming the protein-binding motif to be of width w, MDscan enumerates each non-redundant w-mer (seed) that appears in both strands of the top t sequences and searches for all w-mers in the top t sequences with at least m base pairs matching the seed (called m-matches). The m is determined so that the chance that two randomly generated w-mers are m-matches of each other is smaller than a certain threshold, such as 2%. For each seed, MDscan finds all the m-matches in the top t sequences and uses them to form a motif weight matrix. If the expected number of bases per motif site in the top sequences can be estimated, the following approximate maximum a posteriori scoring function can be used to evaluate a matrix:

$$\frac{x_m}{w} \times \left[\sum_{i=1}^{w} \sum_{j=A}^{T} p_{i,j} \log p_{i,j} - \frac{1}{x_m} \sum_{\text{all segments}} \log(p_0(s)) - \log(\text{expected bases/sites}) \right],$$

where x_m is the number of m-matches aligned in the motif, $p_{i,j}$ is the frequency of nucleotide j at position i of the motif matrix and $p_0(s)$ is the probability of generating the m-match s from the background model. A Markov background model is used and estimated from all the intergenic regions of a genome. When the expected number of sites in the top sequences is unknown, the motif matrix can also be evaluated by:

$$\frac{\log(x_m)}{w} \times \left[\sum_{i=1}^{w} \sum_{j=A}^{T} p_{i,j} \log p_{i,j} - \frac{1}{x_m} \sum_{\text{all segments}} \log(p_0(s)) \right].$$

After computing the scores for all the w-mer motifs established in this step, the highest 10–50 “seed” candidate motifs are retained for further improvement in the next step. In the motif improvement step, every retained candidate motif weight matrix is used to scan all the w-mers in the remaining sequences. A new w-mer is added into a candidate weight matrix if and only if the motif score of that matrix is increased. Each candidate motif is further refined by re-examining all the segments that are already included in the motif matrix during the updating step. A segment is removed from the matrix if doing so increases the motif score. The aligned segments for each motif usually stabilize within ten refinement iterations. MDscan reports the highest-scoring candidate motifs as the protein-DNA interaction motif. With minor modifications, MDScan can be also applied to the TF-DNA affinities measured by protein binding microarrays to discover TF binding motifs.

Jensen et al. [13] provided a perspective of the motif finding problem from the viewpoint of optimizing a scoring function. Several scoring functions were derived based on both Bayesian and non-Bayesian arguments, and compared together with the scoring function used by MDScan. Simulation analyses and a real-data example showed that scoring functions resulting from proper posterior distributions, or approximations to such distributions, showed the best performance and can be used to improve upon existing motif-finding programs.

3.4 Predictive Models for TF-DNA Interaction

As introduced in the previous section, a widely used model for characterizing the common sequence pattern of a set of TFBSs is the PSWM. Although statistical models being employed are already quite intricate, the predictive accuracies of these PSWM-based methods for TFBSs and CRMs are still not fully satisfactory. It is extremely difficult to build more complicated generative models that are both scientifically and statistically sound. First, the data used to estimate model parameters are limited to only several to several tens of known binding sites. With this little information, it is hardly feasible to fit a complicated generative model that is useful for prediction. Second, the detailed mechanism of TF-DNA interaction, which is likely gene-dependent, has not been understood well enough so as to suggest faithful quantitative models. For example, it is well-known that nucleosome occupancy and histone modifications play important roles in gene regulation in eukaryotes, but it is not clear how to incorporate them into a TF-DNA binding model. The predictive modeling approach described in this section explores from a different angle to address such complications.

3.4.1 Joint Analysis of Sequence Motifs and Expression Microarrays

A highly successful tactic for TF motif discoveries is to cluster genes based on their expression profiles, and search for enriched sequence patterns in the sequences upstream of tightly clustered genes. When noise is introduced into the cluster through spurious correlations, however, such an approach may result in many false positives.

Bussemaker et al. [3] proposed a novel method for TFBM discovery via the association of gene expression values with abundance of certain oligomers. They first conducted word enumeration and then used regression to check whether the genes whose upstream sequences contain a set of words have significant changes in their expression.

Conlon et al. [5] presented an alternative approach, MotifRegressor, operating under the assumption that, in response to a given biological condition, the effect of a TF binding motif is approximately linear, the strongest among genes with the most dramatic increase or decrease in mRNA expression. The method combines the advantages of matrix-based motif finding and oligomer motif-expression analysis, resulting in a high sensitivity and specificity. MDscan introduced in the previous section is first used to generate a large set of candidate motifs that are enriched (maybe only slightly; it is not necessary to be stringent here) in the promoter regions (DNA sequences) of genes with the highest fold change in mRNA level relative to a control condition. How well the upstream sequence of a gene g matches a motif m, in terms of both degree of matching and number of sites, is determined by the following likelihood-ratio function:

$$S_{m,g} = \log_2 \left[\sum_{x \in X_{m,g}} P(x \text{ from } \theta_m)/P(x \text{ from } \theta_0) \right],$$

where θ_m is the probability matrix of width, θ_0 represents the third-order Markov model estimated from all of the intergenic sequences (or all the sequences in the given dataset), and $X_{m,g}$ is the set of all w-mers in the upstream sequence of gene g. For each motif reported by MDscan, MotifRegressor first fits the simple linear regression:

$$Y_g = \alpha + \beta_m S_{m,g} + \epsilon_g,$$

where Y_g is the $\log_2$-expression value of gene g and ϵ_g is the gene-specific error term. The baseline expression α and the regression coefficient β_m will be estimated from the data. A significantly positive or negative β_m indicates that motif m (and its corresponding binding TF) is very likely associated with the observed gene expression changes.

The candidate motifs with significant p-values ($P \leq 0.01$) for the simple linear regression coefficient β_m are retained and used by the stepwise regression procedure to fit a multiple regression model:

$$Y_g = \alpha + \sum_{m=1}^{M} \beta_m S_{m,g} + \epsilon_g.$$

Stepwise regression begins with only the intercept term, and adds at each step the motif that gives the largest reduction in residual error. After adding each new motif m, the model is checked to remove the ones whose effects have been sufficiently explained by m. The final model is reached when no motif can be added with a statistically significant coefficient. Since the above procedure involves three steps: motif finding, simple linear regression, and stepwise regression, it is infeasible to compute statistical significance of the final regression result analytically. A permutation-type testing procedure was implemented instead.

There are some additional challenges that require for more sophisticated methods than linear regression, such as nonlinear relationship, large number of possible variables, and the generation of meaningful variables. Machine learning strategies like the boosting method [12] have been used to build more accurate binding model for transcription factors. Zhong et al. [29] proposed to use regularized sliced inverse regression (RSIR) to select relevant motifs. RSIR assumes that gene i's transcription rate y_i and its sequence motif scores $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,M})^T$ are related as:

$$y_i = f(\boldsymbol{\beta}_1^T \mathbf{x}_i, \ldots, \boldsymbol{\beta}_k^T \mathbf{x}_i, \epsilon_i),$$

where $f(\cdot)$ is an unknown (and possibly nonlinear) function, $\boldsymbol{\beta}_l = (\beta_{l,1}, \ldots, \beta_{l,M})^T$, are vectors of linear coefficients, and ϵ_i represents the noise. The number k is called the dimension of the model. A linear regression model is a special one-dimensional case of this model. RSIR estimates both k and the $\boldsymbol{\beta}_l$ values without estimating $f(\cdot)$. Since many entries of the $\beta_{l,j}$ values are close to zero, which implies that the corresponding motif scores contribute very little, only those motifs whose coefficient $\beta_{l,j}$ is significantly nonzero are retained. Compared with linear regression, RSIR is efficient in computation, very stable for data with high dimensionality and high collinearity, and improves motif detection sensitivities and specificities by avoiding inappropriate model specifications.

3.4.2 Modeling TF-DNA Interaction and Genome-wide Occupancy Data

To give a physical view of the predictive modeling approach for TF-DNA binding, consider a reversible reaction of the TF to a short piece of DNA schematically represented by

$$\text{TF} + \text{DNA} \rightleftharpoons \text{TF-DNA}.$$

The rates depend on the DNA sequence s. Let $K_{bind}(s)$ and $K_{diss}(s)$ be the sequence-dependent rate constants for TF binding and for TF dissociation, respectively. If the binding free energy of a TF to a short stretch of DNA with sequence s is $G(s)$, then

$$\frac{K_{bind}(s)}{K_{diss}(s)} = K \exp(-\beta G(s)),$$

where $\beta = 1/k_B T$, with k_B being the Boltzmann constant and K a constant. When such a sequence is in a solution containing the transcription factor with the concentration v_{TF}, the equilibrium probability of it being bound to a TF molecule is

$$\begin{aligned} p(s) &= \frac{[\text{TF-DNA}]}{[\text{TF-DNA}] + [\text{DNA}]} = \frac{K_{bind}(s) v_{TF}}{K_{bind}(s) v_{TF} + K_{diss}(s)} \\ &= \frac{K \exp(-\beta G(s)) v_{TF}}{K \exp(-\beta G(s)) v_{TF} + 1}, \end{aligned}$$

which can be rewritten in the form of an inverse logit function:

$$p(s) = \frac{1}{\exp\{(G(s) - \mu)/k_B T\} + 1},$$

recognized as the Fermi-Dirac distribution, where μ is the chemical potential set by the TF concentration $\mu = k_B T \ln(K v_{TF})$. The Fermi-Dirac form of binding probability tells us that a sequence with binding energy well below the chemical potential (which depends on the factor concentration) is almost always bound to a factor. On the other hand, if the binding energy is well above the chemical potential, the sequence is rarely bound, with the binding probability approximated by $\exp\{(G(s) - \mu)/k_B T\}$. Djordjevic et al. [6] constructed energy matrix for the DNA binding proteins in E.coli from the known binding sites based on this model.

Recent technologies such as and ChIP-array and protein binding microarrays (PBM) provide direct and quantitative information about the TF occupancy of large genomic regions. For each DNA segment s there are two microarray intensities. The test intensity I_s^{test} is equal to a background intensity α_{test} times a term that is proportional to the approximate binding probability (occupancy) $\exp\{(G(s) - \mu)/k_B T\}$ by the TF, either because the amount of TF bound to the probe contributes directly to the signal intensity (PBM) or because it determines the proportion at which an immunoprecipitated TF-DNA fragment is present in the sample (ChIP-array). The control intensity $I_s^{control}$ is only the result of background signal $\alpha_{control}$. Thus,

$$\frac{I_s^{test}}{I_s^{control}} \propto \frac{\alpha_{test}}{\alpha_{control}} \exp\{(G(s) - \mu)/k_B T\}.$$

Based on this model with a simple assumption that each base within a contiguous DNA string s contributes independently to the overall Gibbs free energy $G(s)$, Foat et al. [7] developed the MatrixREDUCE algorithm, which uses ChIP-array data for a TF and associated nucleotide sequences to discover the sequence specific binding affinity of the TF. More recently, Kinney et al. [14] presented a likelihood-based approach for inferring physical models of TF-DNA binding energy from the data produced by PBM and ChIP-array data.

The observed intensity ratio gives a noisy measure of the enrichment of the TF-DNA complex. Assuming an additive error ϵ on the logarithmic scale, we obtain:

$$\text{log intensity ratio (LIR)} = \log(I_s^{test}/I_s^{control}) = \beta_0 + \beta_1 G(s) + \epsilon,$$

where $G(s)$ is the energy score of binding site for the TF in the sequence. Suppose that Y is the observed log-ChIP-intensity and that $G(s)$ can be written as a function f of the extracted sequence features $\mathbf{x} = X_1, \ldots, X_p$, and then the model becomes:

$$Y = f(\mathbf{X}) + \epsilon.$$

The model above serves as the basis for the predictive modeling framework.

3.4.3 Selecting Sequence Features to Predict TF-DNA Interactions

By treating gene expression or ChIP-array intensity values as response variables and a set of candidate motifs (in the form of PSWMs) and/or other sequence features as potential predictors, predictive modeling approaches infer a statistical relationship between genomic sequences and gene expression or ChIP-binding intensities through a regression framework, and influential sequence features are identified by variable selection. Zhou and Liu [31] presented a systematic study of predictive modeling approaches to the TF-DNA binding problem and examined a few contemporary statistical learning methods for their power in expression/ChIP-intensity prediction and in selection of relevant sequence features.

A critical step that determines whether the method will be ultimately successful for a real problem is to extract sequence features, that is, to transform the sequence data into vectors of numerical values. Three categories of sequence features of a DNA sequence have been entertained: the generic, the background, and the motif features. Generic features include the GC content, the average conservation score of a sequence and the sequence length. Background features count the occurrences of all the 2-mers and 3-mers in a DNA sequence. Motif features of a DNA sequence are derived from a precompiled set of TF-binding motifs, each represented by a PSWM. The compiled set includes both known motifs from TF databases and new motifs found from the positive ChIP sequences in the data set of interest using a de novo motif search tool. A heterogeneous (i.e., segmented) Markov background model was fitted for a sequence to account for the heterogeneous nature of genomic sequences. Intuitively, this model assumes that the sequence in consideration can be segmented into an unknown number of pieces and, within each piece, the nucleotides follow a homogeneous first-order Markov chain, and finally a motif score for a sequence was defined similar to the likelihood-ratio function used in MotifRegressor.

Zhou and Liu [31] examined a few state-of-the-art learning methods including stepwise linear regression, neural networks, multivariate adaptive regression splines (MARS) [9], support vector machines (SVM) [25], boosting [8], and Bayesian additive regression trees (BART) [4]. These methods are applied to both simulated datasets and two whole-genome ChIP-array datasets. They found that, with proper learning methods, predictive modeling approaches can significantly improve the predictive power and identify more biologically interesting features, such as TF-TF interactions. A special attention is paid to the Bayesian learning strategy BART, which was demonstrated to have the best overall performance.

In contrast to PSWM-based generative models constructed from biophysics heuristics, predictive modeling approaches aim to learn from the data a flexible model to approximate the conditional distribution of the response variable given the potential predictors. In doing so, they are able to not only pick up relevant sequence features, but also incorporate other genomic features such as nucleosome occupancy and histone modification markers.

3.4.4 Integrating Epigenetic Features to Predict TF-DNA Interactions

Gene activities in eukaryotic cells are regulated by a concerted action of and interaction between transcription factors and chromatin structure. The basic repeating unit of chromatin is the nucleosome, an octamer containing two copies each of four core histone proteins. Although relatively little is known about the hierarchical organization of chromatin, nucleosome is now believed to have a role in regulating transcription by controlling access of transcription factors to the genome. It has been shown that regulatory elements such as promoters and enhancers are associated with distinct chromatin signatures and conversely such chromatin signatures could be used to predict the regulatory elements. Narlikar et al. [22] described a motif discovery algorithm that employs an informative prior over DNA sequence positions based on a discriminative view of nucleosome occupancy. When a Gibbs sampling algorithm is applied to yeast sequence-sets identified by ChIP-array, the correct motif is found in 52% more cases with informative prior than with the commonly used uniform prior. This improvement is expected to be more dramatic as high-resolution genome-wide experimental nucleosome occupancy data becomes increasingly available.

While nucleosome occupancy in promoter regions typically occludes transcription factor binding, thereby repressing global gene expression, the role of histone modification is more complex. Histone tails can be modified in various ways, including acetylation, methylation, phosphorylation, and ubiquitination. Even the regulatory role of histone acetylation, the best characterized modification to date, is still not fully understood. It is thus important to assess the global impact of histone acetylation on gene expression, especially the combinatory effect of histone acetylation sites and the confounding effect of sequence dependent gene regulation and histone occupancy. Yuan et al. [28] proposed a statistical approach to to evaluate the regulatory effect of histone acetylation by combining available genome-wide data from histone acetylation, nucleosome occupancy and transcriptional rate. The combined transcriptional control by TFBMs, nucleosome occupancy, and histone acetylation is modeled as follows:

$$y_i = \alpha + \sum_j \beta_j x_{i,j} + \sum_k \eta_k z_{i,k} + \delta w_i + \epsilon_i,$$

where y_i is the transcription rate of gene i, the $x_{i,j}$ values are the three histone acetylation levels (corresponding to H_3K_9, H_3K_{14} and H_4, respectively), the $z_{i,k}$ values are the corresponding scores to the selected motifs, and w_i is the nucleosome occupancy level. A simple regression of transcription rates against histone acetylation without considering any other factors gave an R^2 of 0.2049, implying that about 20% of the variation of the transcription rates is attributable to histone acetylation. In contrast, the comprehensive model with all the variables bumped up the R^2 to 0.3535, indicating that the histone acetylation does have a significant effect on the transcription rate, although not as high as that in the naïve model.

Recent mapping of histone modifications using ChIP-array or ChIP-seq technologies provides an opportunity for predicting TFBSs using an alternative approach. Won et al. [26] proposed an integrated approach that combines sequence information and chromatin signatures to predict binding sites of individual TFs. The proposed method integrates the sequence information and ChIP-seq signals of histone modifications at promoters or enhancers using a hidden Markov model (HMM) that was designed to capture characteristic patterns of these signals.

3.5 Summary

This chapter has reviewed a few statistical models used in predicting TF-DNA interactions based on genomic features, the corresponding statistical formulations, and related computational strategies. Two main modeling strategies have been discussed here. For a generative modeling approach, separate statistical models are fitted to TF-bound (positive) and background (negative) sequences, and then the posterior odds ratio or the likelihood ratio is applied to construct prediction rules. In contrast, a predictive modeling approach targets at prediction by modeling directly the conditional distribution of TF-binding given extensively extracted sequence features. These two approaches have their own respective advantages. If the underlying data generation process is unclear or difficult to model, predictive approaches have the advantage to construct an informative conditional distribution from the training data. On the other hand, generative models are usually built with more explicit assumptions that help us understand the underlying science and can capture key characteristics of a biological or physical system.

Our main goal here is to introduce several effective statistical tools for combing high-throughput experimental data (expression microarray, ChIP-array/seq, protein binding microarray) with genomic sequence and epigenetic data to tackle the protein-DNA binding problem. Along this direction, we have introduced a general framework to explore and characterize potentially influential factors. The finding that an integrated model can significantly improve the predictive power indicates potentially important yet less understood roles different genomic factors play in TF-DNA interactions. With the rapid accumulation of large-scale genomic data, we believe that more flexible statistical methods integrating generative and predictive models will be very useful for studying a large class of biological problems including TF-DNA interaction.

References

1. Bailey, T. L., & Elkan, C. (1994). Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proceedings of the second international conference on intelligent systems for molecular biology* (pp. 28–36). Menlo Park, California: AAAI Press.
2. Berger, M. F., Philippakis, A. A., Qureshi, A., He, F. S., Estep, P. W., & Bulyk, M. L. (2006). Compact, universal DNA microarrays to comprehensively determine transcription-factor binding site specificities. *Nature Biotechnology*, *24(11)*, 1429–1435
3. Bussemaker, H. J., Li, H., & Siggia, E. D. (2001). Regulatory element detection using correlation with expression. *Nature Genetics*, *27*, 167–174.
4. Chipman, H. A., George, E. I., & McCulloch, R. E. (2007). Bayesian ensemble learning. In B. Scholkopf, J. Platt, & T. Hoffman (Eds.), *Neural information processing systems, 19*. Cambridge, MA: MIT Press.
5. Conlon, E. M., Liu, X. S., Lieb, J. D., & Liu, J. S. (2001). Integrating regulatory motif discovery and genome-wide expression analysis. *Proceedings of the National Academy of Science United States of America*, *100*, 3339–3344.
6. Djordjevic, M., Sengupta, A. M., & Shraiman, B. I. (2003). A biophysical approach to transcription factor binding site discovery. *Genome Research*, *13*, 2381–2390.
7. Foat, B. C., Houshmandi, S. S., Olivas, W. M., & Bussemaker, H. J. (2005). Profiling condition-specific, genome-wide regulation of mRNA stability in yeast. *Proceedings of the National Academy of Science United States of America*, *102*, 17675–17680.
8. Freund, Y., & Schapire, R. (1997). A decision-theoretical generalization of online learning and an application to boosting. *Journal of Computer and System Science*, *55*, 119–139.
9. Friedman, J. H. (1991). Multivariate adaptive regression splines. *Annals of Statistics*, *19*, 1–67.
10. Gupta, M., & Liu, J. S. (2005). De-novo cis-regulatory module elicitation for eukaryotic genomes. *Proceedings of the National Academy of Science United States of America*, *102*, 7079–7084.
11. Hertz, G. Z., Hartzell, G. W., & Stormo, G. D. (1990). Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Bioinformatics*, *6*, 81–92.

12. Hong, P., Liu, X. S., Zhou, Q., Lu, X., Liu, J. S., & Wong, W. H. (2005). A boosting approach for motif modeling using ChIP-chip data. *Bioinformatics*, *21*, 2636–2643.
13. Jensen, S. T., Liu, X. S., Zhou, Q., & Liu, J. S. (2004) Computational discovery of gene regulatory binding motifs: A bayesian perspective. *Statistical Science*, *19*, 188–204.
14. Kinney, J. B., Tkacik, G., & Callan, C. G., Jr. (2007). Precise physical models of protein-DNA interaction from high-throughput data. *Proceedings of the National Academy of Science United States of America*, *104*, 501–506.
15. Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F., & Wootton, J. C. (1993). Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science*, *262*, 208–214.
16. Lee, W., Tillo, D., Bray, N., Morse, R. H., Davis, R. W., Hughes, T. R., et al. (2007). A high-resolution atlas of nucleosome occupancy in yeast. *Nature Genetics*, *39*, 1235–1244.
17. Liang, F., & Wong, W. H. (2002). Evolutionary Monte Carlo: Applications to Cp model sampling and change point problem. *Statistica Sinica*, *10*, 317–342.
18. Liu, J.S., & Lawrence, C.E. (1999). Bayesian inference on biopolymer models. *Bioinformatics*, *15*, 38–52.
19. Liu, J. S., Neuwald, A. F., & Lawrence, C. E. (1995). Bayesian models for multiple local sequence alignment and Gibbs sampling strategies. *Journal of the American Statistical Association*, *90*, 1156–1170.
20. Liu, X. S., Brutlag, D. L., & Liu, J. S. (2002). An algorithm for finding protein-DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments. *Nature Biotechnology*, *20*, 835–839.
21. McCue, L. A., Thompson, W., Carmack, C. S., Ryan, M. P., Liu, J. S., Derbyshire, V., & Lawrence, C. E. (2001). Phylogenetic footprinting of transcription factor binding sites in proteobacterial genomes. *Nucleic Acids Research*, *29*, 774–782.
22. Narlikar, L., Gordân, R., & Hartemink, A. J. (2007). A nucleosome-guided map of transcription factor binding sites in yeast. *PLoS Computational Biology*, *3(11)*, e215
23. Sinha, S., & Tompa, M. (2002). Discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Research*, *30*, 5549–5560.
24. Thompson, W., Palumbo, M. J., Wasserman, W. W., Liu, J. S., & Lawrence, C. E. (2004). Decoding human regulatory circuits. *Genome Research*, *10*, 1967–1974.
25. Vapnik, V. (1998). *The nature of statistical learning theory* (2nd ed.). New York: Springer.
26. Won, K. J., Ren, B., & Wang, W. (2010). Genome-wide prediction of transcription factor binding sites using an integrated model. *Genome Biology*, *11*, R7.
27. Yuan, G. C., Liu, Y. J., Dion, D. F., Slack, M. D., Wu, L. F., Altschuler, S. J., et al. (2005). Genome-scale identification of nucleosome positions in S. cerevisiae. *Science*, *309*, 626–630.
28. Yuan, G. C., Ma, P., Zhong, W., & Liu, J. S. (2006). Statistical assessment of the global regulatory role of histone acetylation in Saccharomyces cerevisiae. *Genome Biology*, *7*, R70.
29. Zhong, W., Zeng, P., Ma, P., Liu, J. S., & Zhu, Y. (2005). RSIR: regularized sliced inverse regression for motif discovery. *Bioinformatics*, *21*, 4169–4175.
30. Zhou, Q., & Liu, J. S. (2004). Modeling within-motif dependence for transcription factor binding site predictions. *Bioinformatics*, *20*, 909–916.
31. Zhou, Q., & Liu, J. S. (2008). Extracting sequence features to predict protein-DNA interactions: A comparative study. *Nucleic Acids Research*, *36*, 4137–4148.
32. Zhou, Q., & Wong, W. H. (2004). CisModule: De novo discovery of cis-regulatory modules by hierarchical mixture modeling. *Proceedings of the National Academy of Science United States of America*, *101*, 12114–12119.

Chapter 4
Computational Promoter Prediction in a Vertebrate Genome

Michael Q. Zhang

Abstract Computational prediction of vertebrate gene promoters from genomic DNA sequences is one of the most difficult problems in computational genomics, but it is essential for understanding genome organization, improving gene annotation and for further comprehensive studies of gene expression and regulation networks. The advent of new genomic technologies has ushered forth the era of deeper understanding of molecular biology at systems level, more accurate and diverse large-scale molecular data have been fueling the development of new predictive methods and computational tools in this rapidly moving field. In this chapter, I will give an introduction on structure and function of promoters in typical vertebrate genes, as well as experimental methods for determining them. I then describe generic statistical methods for promoter prediction and a few computational approaches as examples. I will further review and update on more recent advances in promoter prediction methodologies and give a future prospect in the conclusion.

4.1 Biological Background on Promoter Structure and Function

In this chapter, we mainly focus on protein coding gene promoters. We will briefly mention miRNA gene promoters in the end because they have become increasingly important and majority of them are also transcribed by polymerase II.

According to the central dogma, during gene expression process, each protein gene in the genome must be first transcribed into pre-mRNA transcripts before they can be further processed into mRNAs and transported out of the nucleus into

M.Q. Zhang
The University of Texas at Dallas, 800 West Cambell Rd, RL11, Richardson, TX 75080-3021,
1-972-883-2523
and
Tsinghua University, Beijing, 100084, China
e-mail: MichaelQiweiZhang@gmail.com

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_4,
© Springer-Verlag Berlin Heidelberg 2011

cytoplasm for protein translation. The first base pair in the DNA corresponding to the beginning of a pre-mRNA transcript is called TSS (Transcriptional Start Site), the promoter that can "drive" the transcription of its target gene is loosely defined by a piece of DNA region ($500 \sim 2$ kb) around the TSS. We need to make three remarks: (1) Since a typical promoter can drive transcription at multiple TSSs (strictly speaking, each promoter can drive a distribution of TSSs), for simplicity, when we say a TSS, it should generally be understood as a typical or the median of the TSS distribution. (2) Each gene can have multiple promoters (actually most of the human genes do), different promoters must have minimum inter-distance ($\sim$500 bp in human) apart and we will briefly discuss alternative promoter prediction in the end. (3) Some people also refer more extended region ($2 \sim 10$ kb) as promoters that also contain the enhancers that can activate (or de-repress) the promoter. Since the enhancers that are required for precise in vivo gene expression pattern can be even farther away and in both upstream and downstream of TSS, computationally it is convenient (and often necessary) not to include any such distal enhancers as part of promoter definition. Even with 2 kb definition, a promoter may already contain some functional enhancers. People often refer (200 bp–1 kb) promoters as the proximal promoters.

4.2 Core-Promoter Structure and Function

Computationally, it often refers ($\sim$80–100 bp) region centered at a TSS as the core-promoter. It can be defined operationally in vitro by a minimum piece of DNA that is required for the assembly of the pre-initiation complex (PIC) and can drive a reporter gene transcription specifically from the TSS at a basal level. Since promoter is the cis-control and regulatory region for the target gene, the most distinguishing property is that it contains many transcription factor binding sites (TFBSs). In particular, a core-promoter is often "packed" by such functional elements, many of which are bounded by general transcription factors (GTFs). Some core-promoter elements are well-known, including the TATA box, the initiation (Inr), the downstream promoter element (DPE), the TFIIB recognition elements (BRE), the motif-ten-element (MTE), downstream core element (DCE), XCPE1 and XCPE2, the latter ones are not present in a large number of genes. Each such element has a specific, albeit degenerate, DNA sequence pattern (motif), together they form the basis for promoter recognition molecularly through protein-DNA interactions. The function of core-promoter is to recruit polymerase II at TSS and to initiate target gene expression in response to regulatory signals.

4.3 Typical Experimental Methods for Identification of Promoters

4.3.1 Nuclease S1 Protection and Primer Extension Assays

These methods are considered as highly accurate experimental assays for TSS mapping. The endonuclease S1 is an enzyme from the mold *Aspergillus oryzae* which cleaves single-stranded RNA and DNA but not double-stranded molecules. In order to map the TSS for a gene by nuclease S1 protection, a genomic DNA clone suspected of containing the start site is required. The DNA clone is then digested with a suitable restriction endonuclease to generate a fragment that is expected to contain the TSS. As shown in Fig. 4.1a, hybridization to the cognate mRNA and S1 nuclease digestion defines the distance of the TSS from the unlabeled end of the restriction fragment. If more precise localization is required, the labeled DNA fragment in the heteroduplex can be sequenced.

The primer extension method is very similar to the nuclease S1 protection method. In this case, the chosen restriction fragment must be shorter than the mRNA

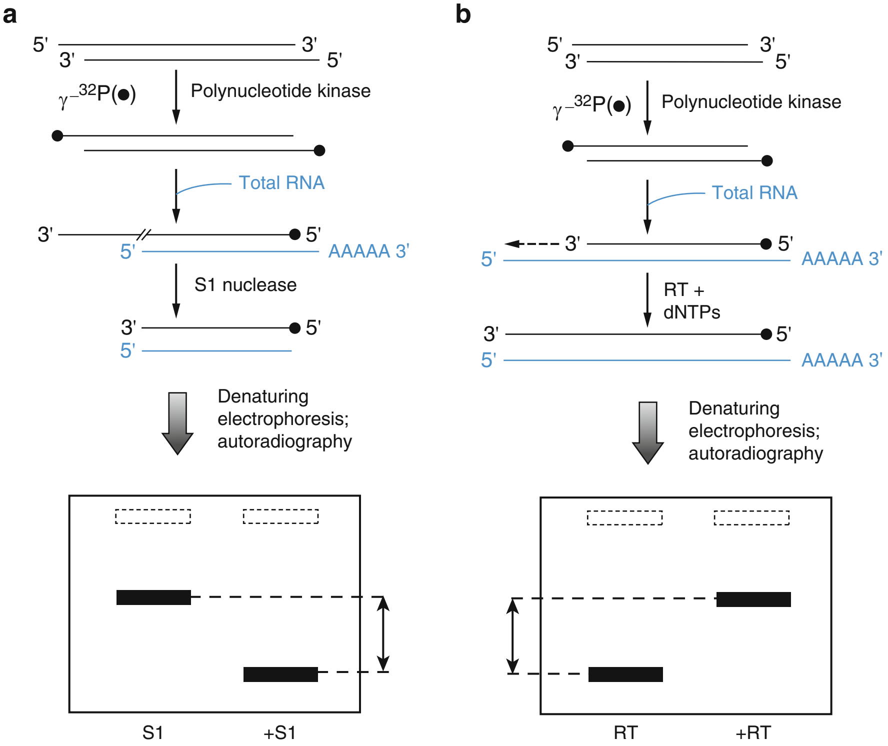


Fig. 4.1 Nuclease S1 protection (**a**) and primer extension (**b**)

and the overhang is filled in using reverse transcriptase (Fig. 4.1b). Again a more accurate location is possible by sequencing the labeled DNA strand.

4.3.2 5′-RACE-PCR and Cage -Tagging

One popular method of obtaining cDNA end sequences is the RACE (rapid amplification of cDNA ends) technique. RACE-PCR is an anchor PCR modification of RT-PCR. The rationale is to amplify sequences between a single previously characterized region in the mRNA (cDNA) and an anchor sequence that is coupled to the 5′ or the 3′ end. A primer is designed from the known internal sequence and the second primer is selected from the relevant anchor sequence (see Fig. 4.2, similarly, 3′ RACE-PCR may be used to map 3′UTR end of a mRNA).

CAGE (Cap Analysis Gene Expression) relies on a cap-trapper system to capture full-length RNAs while avoiding rRNA and tRNA transcripts. First, an oligodT primer is used to reverse-transcribe poly-A terminated RNAs. Alternatively, a

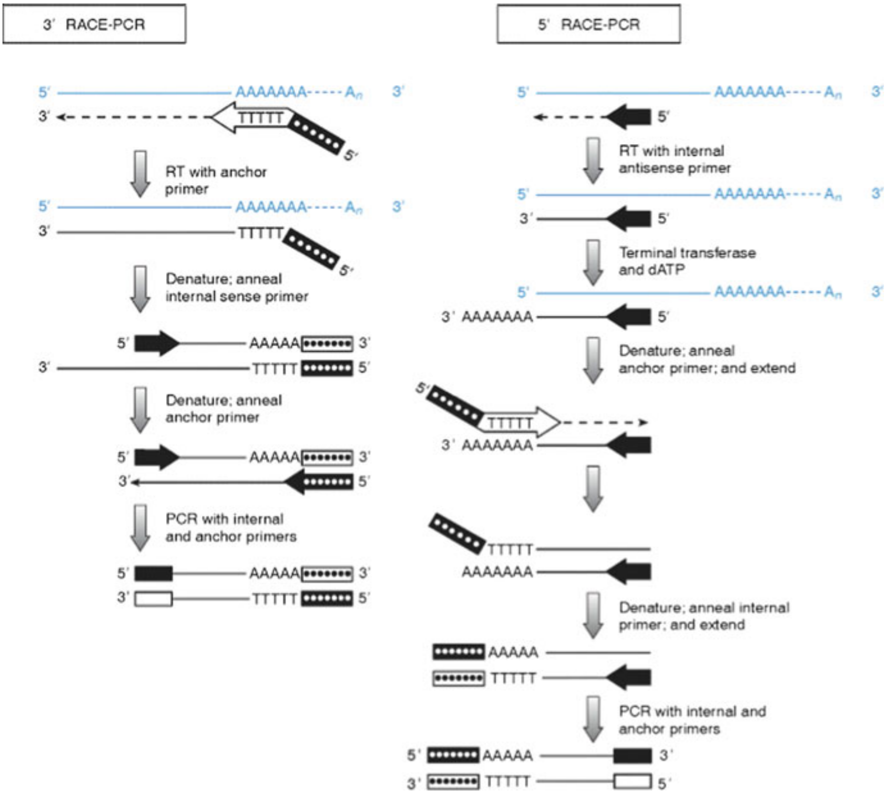
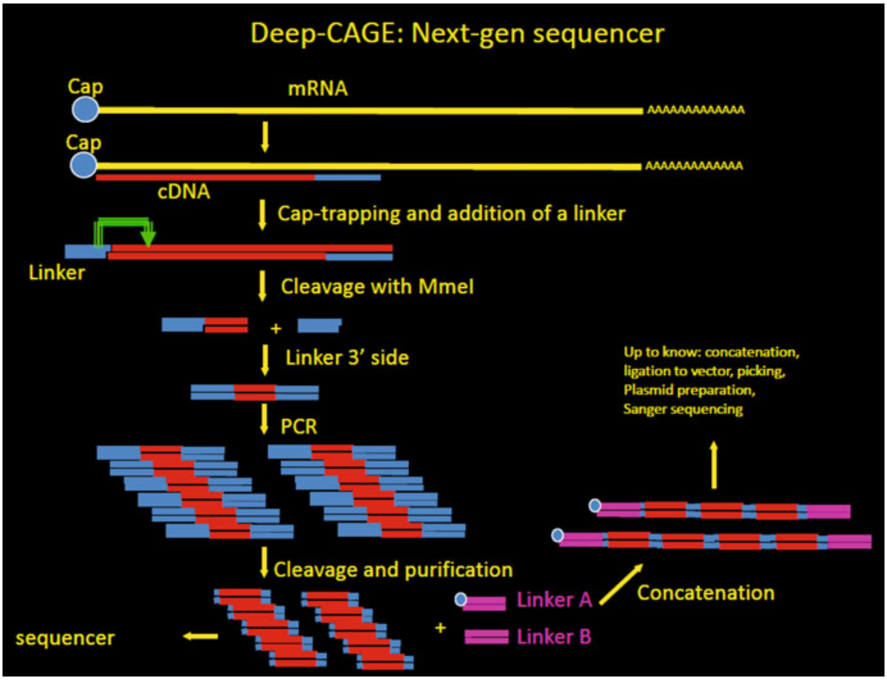

3′ RACE-PCR
5′ AAAAAAA-----An 3′
3′ TTTTT 5′
RT with anchor primer
Denature; anneal internal sense primer
AAAAA
TTTTT
Denature; anneal anchor primer
PCR with internal and anchor primers
5′ RACE-PCR
RT with internal antisense primer
Terminal transferase and dATP
3′ AAAAAAA
Denature; anneal anchor primer; and extend
TTTTT
Denature; anneal internal primer; and extend
PCR with internal and anchor primers

Fig. 4.2 RACE-PCR methods



Deep-CAGE: Next-gen sequencer

Cap

mRNA

AAAAAAAAAAAA

Cap

AAAAAAAAAAAA

cDNA

Cap-trapping and addition of a linker

Linker

Cleavage with MmeI

+

Linker 3′ side

PCR

Up to know: concatenation,
ligation to vector, picking,
Plasmid preparation,
Sanger sequencing

Cleavage and purification

Linker A

Concatenation

sequencer

+

Linker B

Fig. 4.3 Deep-CAGE sequencing (Dr. Piero Carninci at Riken, Japan, gave me one of his slide from his talk)

random primer can be used for RNAs without a poly-A tail, which may constitute almost half of the transcriptome. RNA/DNA double-stranded hybrids that contain a mature mRNA are selected by biotinylating their 5′ cap structure, allowing capture by streptavidin-coated magnetic beads. Ligation of a linker sequence containing an MmeI recognition site to the 5′ end of the full-length cDNA creates a restriction site about 20 nucleotides downstream, producing a short CAGE tag starting at the 5′ end of mRNAs. These tags are amplified by PCR. Traditionally such amplified short tags are concatenated (like SAGE tags) for conventional Sanger or 454 sequencing, more recently they are directly sequenced by next-generation sequencer, such as Illumina/Solexa sequencer.

4.3.3 ChIP-chip and ChIP-seq

TSS may also be approximately localized by mapping promoter binding proteins (e.g. polII itself, TAF1, H3K4me3). **ChIP-on-chip** (also known as **ChIP-chip**) is a technique that combines chromatin immunoprecipitation ("*ChIP*") with microarray technology ("*chip*"). Like regular ChIP, ChIP-on-chip is used to investigate interactions between proteins and DNA in vivo. Specifically, it allows the identification of the cistrome, sum of binding sites, for DNA-binding proteins on a genome-wide basis. Whole-genome analysis can be performed to determine the locations of binding sites for almost any protein of interest. As the name of the technique suggests, such proteins are generally those operating in the context of chromatin. The most prominent representatives of this class are transcription factors, replication-related proteins, like ORC, histones, their variants, and histone modifications. The goal of ChIP-on-chip is to localize protein binding sites which may help in identifying functional elements in the genome. ChIP-Seq technology is currently seen primarily as an alternative to ChIP-chip which requires a hybridization array. This necessarily introduces some bias, as an array is restricted to a fixed number of probes. Sequencing, by contrast, is thought to have less bias, although the sequencing bias of different sequencing technologies is not yet fully understood. Massively parallel sequence analyses are used in conjunction with whole-genome sequence databases to analyze the interaction pattern of any protein with DNA (see [15]).

4.4 Computational Methods for Promoter Prediction

The simplest method for promoter prediction is to sequence and to map a full length cDNA by alignment to the genome, this has been used as the "gold standard" for bench-marking or validating all *de novo* promoter prediction results in large-scale genomic studies. Since many cDNAs are not known, computational prediction is still valuable in practice; even if we can map experimentally all promoters some day, we still would not understand what determine a functional promoter unless we have mathematical models that can accurately predict it. The *de novo* identification of promoters has been a challenging problem. A two-step approach to promoter recognition and TSS mapping may be necessary: initial identification of a functional promoter in a roughly 2-kilobase (kb) region and further prediction of a TSS within a 50 bp region. The first step is on a larger scale, in which coarse-grained measures such as CpG islands, nucleosome binding, chromatin modification, downstream coding propensity, and transcription factor density should be very useful. The second step is on a finer scale that needs more detailed features, such as distance-specific TFBS correlations, to best discriminate the precise TSS region from its surroundings. Recent advances in experimental technologies provide an ideal situation to revisit this two-step strategy. For example, results from Pol II ChIP-chip or H3K4me3 ChIP-seq analysis can help to focus the search. A core promoter prediction program can be subsequently used to map the TSS finely. With progress in both experimental and computational technologies, the accuracy and resolution of TSS predictions can be further improved by combining these complementary methods.

By now, many computational methods for promoter prediction have been proposed. The underlying principle of these methods is that promoter regions have some characteristic features (both genetic and epigenetic) that make them distinct from nonpromoters. Predictive models using these features to discriminate promoters from nonpromoters are built largely by machine learning or statistical methods and then used to search/scan for new promoters in an input DNA sequence. Many of these methods are reviewed and compared in several recent reports [1, 2, 21, 22]. Although there has been much success in locating the TSSs for CpG-related promoters, the performance for non-CpG-related promoters (about 25% of known genes) is still not satisfactory because of the diverse nature of vertebrate promoter sequences. Here we choose to introduce three machine learning based methods: Eponine [7], ARTS [18] and CoreBoost [23].

4.4.1 Eponine

In Eponine model, the authors generalize a position weight matrix (PWM) by a linear "convolution" over all possible positions with respect to the TSS:

$$F(i, S) = log \sum_{j=-\infty}^{+\infty} P(j) \cdot W(a + i + j; S)$$

where $P(j)$ is a discrete probability distribution; $W(x, S)$ is a PWM, aligning the first column to position, relative to the TSS; and i is the position of the true TSS during training, and is varied along the length of the sequence when scanning a sequence with the trained model. Since linear combination of such positioned matrix scores is equivalent to the well known generalized linear model (GLM) form [14] and such models can be trained using established relevance vector machine (RVM, [19]) procedures, the authors built Eponine classification models by their own implementation of RVM. In order to analyze promoters, it is necessary to explore an extremely large model space of possible weight matrices and position distributions. To facilitate this, the RVM implementation was expanded to allow sampling from this large rule space. The working set is initialized with weight matrices of lengths 4–8, selected at random, and with random, Gaussian position distributions. As rules in the initial working set are discarded by the pruning algorithm, new examples are added. These may be produced by the same logic used to initialize the working set, or represent small changes to existing rules. In Eponine implementation, the allowed sampling moves are as follows: (1) adjust the center position of a distribution; (2) adjust the width parameter of a position distribution; (3) adjust the weights in a DNA weight matrix; (4) construct a new DNA probability distribution at random, then add it as a column at one end (randomly chosen) of a weight matrix; and (5) remove a column from one end of a weight matrix. This gives a hybrid machine-learning approach, combining the RVM with elements of a Monte Carlo sampling approach. Using this hybrid method, a model can be efficiently built from a large space of potential candidate rules.

Eponine models trained on 599 selected mouse full length cDNAs can give simple interpretations (Fig. 4.4). These models consist of four elements: (1) a diffuse preference for CpG enrichment downstream of the start site. This corresponds with the observation that most promoters are associated with a CpG island;

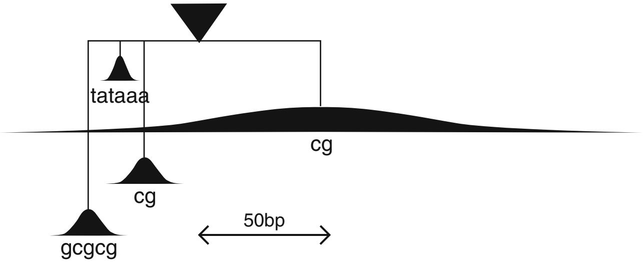


Fig. 4.4 Schematic of Eponine core promoter model, showing the constraint distributions and weight-matrix consensus sequences (the triangle points to the position of TSS) [7]

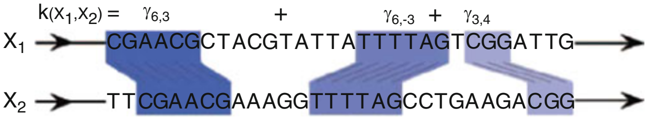

k(X1,X2) = γ6,3 + γ6,-3 + γ3,4
X1 CGAACGCTACGTATTATTTTAGTCGGATTG
X2 TTCGAACGAAAGGTTTTAGCCTGAAGACGG

Fig. 4.5 Given two sequences X1 and X2 of equal length, the WD kernel with shift consists of a weighted sum to which each match in the sequences makes a contribution depending on its length and relative position, where long matches contribute more significantly [18]

(2) a TATAAA motif, with a tightly focused distribution centered at position −30 relative to the transcription start site. This corresponds to the widely reported TATA box and (3 and 4) two GC-rich matrices closely flanking the TATA box.

4.4.2 ARTS

ARTS identifies the TSS through constructing special sub-kernels of Support Vector Machines (SVMs) to combine (both position specific and non-specific) k-mers and other structure features such as twisting angles and stacking energies of the DNA sequence. ARTS consists of five sub-kernels: (1) the extended Weighted Degree kernel with shift (WDs) to capture core promoter sequence elements near a TSS that can have variable lengths and distances to the TSS (Fig. 4.5); (2) a spectrum kernel on a few hundred bps upstream of the TSS to capture certain k-mers that are over- or under- represented (“content sensors”); (3) a similar spectral kernel for the downstream $5'$UTRs as well as further downstream introns and coding regions; (4) two linear kernels to capture three-dimensional DNA structure patterns near the TSS (one for twisting angles and another for stacking energies, both depending on dinucleotides and smoothed by sliding windows).

ARTS uses a linear (equally weighted) combination of these five (normalized) sub-kernels as the final model. All considered kernels correspond to a feature space that can be extremely high dimensional. For instance in the case of the WD kernel on DNA sequences of length 100 with the maximum k-mer length $K = 20$, the corresponding feature space is 10^{14} dimensional (one feature per position and possible k-mer, $1 \leq k \leq K$). The authors solved such technical challenges by developing novel and efficient training and evaluation algorithms using suffix trees and made such computational expensive SVM application practically possible. They claimed ARTS finds about 35% true positives at a false positive rate of 1/1,000, where the other methods (McPromoter, Eponine and FirstEF) find only about 18%. Since ARTS uses only downstream genic sequences as the negative set (non-promoters), and therefore it may get more false-positives from upstream non-genic regions. Furthermore, like Eponine, ARTS does not distinguish if a promoter is CpG-island related or not and it is not clear how ARTS may perform on non-CpG-island related promoters.

4.4.3 CoreBoost

Boosting [6, 9, 13, 17] has been applied successfully to a wide variety of classification problems. It combines many weak classifiers to boost the performance of the final classifier. If we denote the training data as $(x_1, y_1), \ldots, (x_n, y_n)$, which are independently and identically distributed realizations of random variables (X, Y), where X is the feature vector in R^p, Y is the class label from the set $\{-1, 1\}$, and n is the sample size. Denote $f(x)$ a binary classifier:

$$f : R^p \rightarrow \{-1, 1\}$$

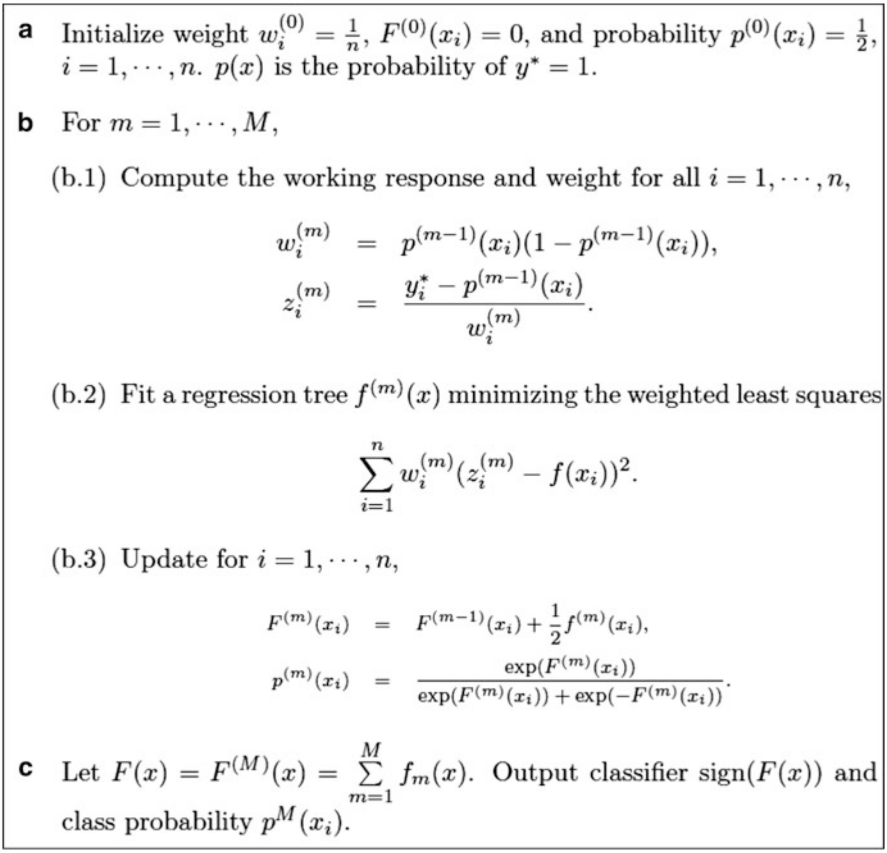
The classifier that minimizes the misclassification risk $P(f(X) \neq Y)$ is called Bayes classifier:

$$f(X) = \begin{cases} 1 & P(Y = 1|X) > P(Y = -1|X) \\ -1 & \text{otherwise.} \end{cases}$$

Denote the ensemble of weak classifiers as follows:

$$F(x) = \sum_{m=1}^{M} c_m f_m(x)$$

where $f_m(x)$ is the mth weak classifier and c_m are constants. At each iteration m, the observations misclassified at the $(m-1)$th iteration are given higher weights for the current iteration. The final ensemble is a weighted majority vote of M weak classifiers (sign $[F(x)]$). CoreBoost uses stumps as weak classifiers. A stump is a special type of decision tree [3] with only two terminal nodes. The boosting algorithm sequentially builds a series of stumps, each trained on reweighted samples. The ensemble of trees has been shown to perform much better than single tree or trees trained independently. For the re-weighting and aggregation, we implement the



a Initialize weight $w_i^{(0)} = \frac{1}{n}$, $F^{(0)}(x_i) = 0$, and probability $p^{(0)}(x_i) = \frac{1}{2}$, $i = 1, \cdots, n$. $p(x)$ is the probability of $y^* = 1$.

b For $m = 1, \cdots, M$,

(b.1) Compute the working response and weight for all $i = 1, \cdots, n$,

$$\begin{aligned} w_i^{(m)} &= p^{(m-1)}(x_i)(1 - p^{(m-1)}(x_i)), \\ z_i^{(m)} &= \frac{y_i^* - p^{(m-1)}(x_i)}{w_i^{(m)}}. \end{aligned}$$

(b.2) Fit a regression tree $f^{(m)}(x)$ minimizing the weighted least squares

$$\sum_{i=1}^{n} w_i^{(m)}(z_i^{(m)} - f(x_i))^2.$$

(b.3) Update for $i = 1, \cdots, n$,

$$\begin{aligned} F^{(m)}(x_i) &= F^{(m-1)}(x_i) + \frac{1}{2} f^{(m)}(x_i), \\ p^{(m)}(x_i) &= \frac{\exp(F^{(m)}(x_i))}{\exp(F^{(m)}(x_i)) + \exp(-F^{(m)}(x_i))}. \end{aligned}$$

c Let $F(x) = F^{(M)}(x) = \sum_{m=1}^{M} f_m(x)$. Output classifier $\text{sign}(F(x))$ and class probability $p^M(x_i)$.

Fig. 4.6 LogitBoost algorithm with trees

LogitBoost algorithm [6], which minimizes the negative of binomial log-likelihood as the loss function. This loss function decreases linearly with $yF(x)$ for misclassified samples and thus is more robust when mislabelled training data are present. The LogitBoost algorithm with decision trees as weak classifiers is outlined in Fig. 4.6. The number M of weak classifiers is determined by using cross-validation. Let $y^* = (y + 1)/2$, taking values from $\{0, 1\}$. LogitBoost directly estimates the posterior class probability:

$$P(Y = 1|X = x) = \frac{e^{F(x)}}{e^{F(x)} + e^{-F(x)}}$$

This is used in the calculation of probability profiles in CoreBoost.

4.4.3.1 Multiclass Classification Using Binary Classifiers

In the application of LogitBoost to the prediction of promoters, there are three classes of outcomes: the promoter (P), its immediate upstream sequence (U), and its immediate downstream sequence (D). Instead of the usual way of combining the upstream and downstream sequences into one class, one can reduce this three-class problem to two binary ones: one comparing the promoter class against the upstream and the other comparing it against the downstream. The reason is that upstream and downstream sequences are very different from each other. Separate classifiers can pick up the most discriminative features for classifying promoters against upstream or downstream sequences. If we denote the following as the probability of Y belonging to the promoter class based on the binary classifier discriminating promoters from the upstream and from the downstream, respectively:

$$p1 = P(Y \in P|X, P, U) \text{ and } p2 = P(Y \in P|X, P, D)$$

The probability p of Y belonging to promoter class in the three-class setting can be calculated as follows:

$$\begin{aligned} P(Y \in P|X, P, U, D) &= \frac{P(Y \in P|X)}{P(Y \in P|X) + P(Y \in U|X) + P(Y \in D|X)} \\ &= \frac{1}{\frac{1}{p1} + \frac{1}{p2} - 1} = \frac{p1p2}{p1 + p2 - p1p2} \end{aligned}$$

CoreBoost uses both immediate upstream and downstream fragments as negative sets and trains separate classifiers for each before combining the two. Extensive experiments showed that better classification accuracy results from use of two binary classifiers rather than one combining the upstream and downstream sequences. It has a false positive rate of $1/5{,}000$ at the sensitivity level of finding 35% true positives. The training sample consists of 300 bp fragments (–250, $+50$), hence it is more localized than ARTS which has training sample of 2 kb fragments $(-1kb, +1kb)$. The ideal application of TSS prediction algorithms is to combine them with gene prediction algorithms and/or with the ChIP-chip or chip-seq PIC mapping data.

4.5 New Advances and Future Challenges

Although much progress has been made in promoter prediction and cis-regulatory motif discovery, false-positives are still the main problem when scanning through the whole genome. Fundamentally this is because the information about chromatin structure is still missing in all our sequenced-based models! Protein-DNA binding specificity is partly determined by the local energetic and partly determined by high-order chromatin structures, which dictates on how much of the genome is accessible to the DNA binding protein [4]. Without knowing which regions of chromatin are open or closed (and to what degree), researchers have to assume the whole genome is accessible for binding, which is obviously wrong and will lead to more false positives (and false negatives because of the extra noises). This is clearly shown by recent genome-wide ChIP-chip/ChIP-seq data as well as DNase I Hypersensitivity mapping data. There is a necessity for higher order prediction algorithms that are capable of predicting chromatin states based upon, perhaps, genome-wide epigenetic measurements, CpG-islands and repeat characteristics in addition to genomic sequences. It is fortunate that such kinds of data are rapidly being generated and the corresponding analysis tools are also coming along. For instance, CoreBoost has recently been further extended to include epigenomic data (e.g. histone modification ChIP-seq data, Fig. 4.7), the new program, called CoreBoostHM [20], has successfully been applied to identify promoters for both protein coding and miRNA genes.

Recent genome-wide studies have revealed much more complex structures of the mammalian core promoters [16], future challenge may be to develop hierarchical models to determine accessible regions in chromatin DNA [5] and to develop a TSS distribution models for different classes of accessible promoters [11]. To further understand the tissue- or developmental regulation of the transcriptome, many more studies are required to predict alternative promoters, bidirectional promoters, antisense transcriptions, and ncRNAs promoters. Hopefully, combining new technologies (e.g. ChIP-seq and RNA-seq) in promoter models will soon help in defining mechanisms that regulate RNA polymerase II transcription in vivo [12]. Finally, comparative modeling will shed more lights on evolution of promoters and their intimate relationship with transposons [8].

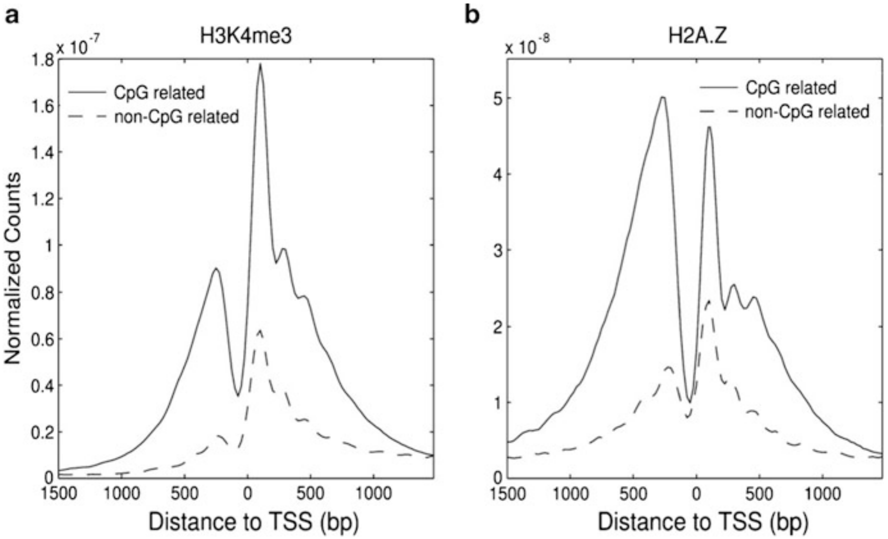


Fig. 4.7 Histone modification signals near TSS for CpG related and non-CpG related promoters [23]

Acknowledgement The author would like to thank the NIH through R01 HG001696 grant.

References

1. Abeel, T., Van de Peer, Y., & Saeys, Y. (2009). Toward a gold standard for promoter prediction evaluation. *Bioinformatics*, *25*(12), i313–i320.
2. Bajic, V. B., Tan, S. L., Suzuki, Y., & Sugano, S. (2004). Promoter prediction analysis on the whole human genome. *Nature Biotechnology*, *22*(11), 1467–1473.
3. Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees*. Belmont, CA: Wadsworth International Group.
4. Buck, M. J., & Lieb, J. D. (2006). A chromatin-mediated mechanism for specification of conditional transcription factor targets. *Nature Genetics*, *38*(12), 1446–1451.
5. Cairns, B. R. (2009). The logic of chromatin architecture and remodeling at promoters. *Nature*, *461*(7261), 193–198.
6. Dettling, M., & Buhlmann, P. (2003). Boosting for tumor classification with gene expression data. *Bioinformatics*, *19*(9), 1061–1069.
7. Down, T. A., & Hubbard, T. J. P. (2002). Computational detection and location of transcription start sites in mammalian genomic DNA. *Genome Research*, *12*(3), 458–461.
8. Faulkner, G. J., & Carninci, P. (2009). Altruistic functions for selfish DNA. *Cell Cycle*, *8*(18), 2895–2900.
9. Freund, Y., & Schapire, R. (1996). Experiments with a new boosting algorithm. *Machine learning: Proceedings of the thirteenth international conference* (pp. 148–156). Italy.

6. Friedman, J., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, *28*(2), 337–407.

11. Frith, M. C., Valen, E., Krogh, A., Hayashizaki, Y., Carninci, P., & Sandelin, A. (2008). A code for transcription initiation in mammalian genomes. *Genome Research*, *18*, 1–12.
12. Fuda, N. J., Ardehali, M. B., & Lis, J. T. (2009). Defining mechanisms that regulate RNA polymerase II transcription in vivo. *Nature*, *461*(7261), 186–192.
13. Kearns, M., & Valiant, L. (1994). Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, *41*(1), 67–95.
14. McCullagh, P., & Nelder, J. A. (1983). *Generalized linear models*. London: Chapman and Hall.
15. Park, P. J. (2009). ChIP–seq: Advantages and challenges of a maturing technology. *Nature Reviews Genetics*, *10*, 669–680.
16. Sandelin, A., Carninci, P., Lenhard, B., Ponjavic, J., Hayashizaki, Y., & Hume, D. (2007). Mammalian RNA polymerase II core promoters: Insights from genome-wide studies. *Nature Reviews Genetics*, *8*(6), 424–436.
17. Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, *5*(2), 197–227.
18. Sonnenburg, S., Zien, A., & Ratsch, G. (2006). ARTS: Accurate recognition of transcription starts in human. *Bioinformatics*, *22*(14), e472–e480.
19. Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning*, *1*, 211–244.
20. Wang, X., Xuan, Z., Zhao, X., Li, Y., & Zhang, M. (2009). High-resolution human core-promoter prediction with CoreBoost_HM. *Genome Research*, *19*(2), 266–275.
21. Zeng, J., Zhu, S., & Yan, H. (2009). Towards accurate human promoter recognition: A review of currently used sequence features and classification methods. *Briefings in Bioinformatics*, *10*(5), 498–508.
22. Zhang, M. Q. (2007). Computational analyses of eukaryotic promoters. *BMC Bioinformatics*, *8*(Suppl. 6), S3.
23. Zhao, X., Xuan, Z., & Zhang, M. (2007). Boosting with stumps for predicting transcription start sites. *Genome Biology*, *8*(2), R17.

Chapter 5
Discovering Influential Variables: A General Computer Intensive Method for Common Genetic Disorders*

Tian Zheng, Herman Chernoff, Inchi Hu, Iuliana Ionita-Laza, and Shaw-Hwa Lo

Abstract We describe a general backward partition method for discovering which of a large number of possible explanatory variables influence a dependent variable Y. This method, based on a variant pioneered by Lo and Zheng, and variations have been used successfully in several biological problems, some of which are discussed here. The problem is an example of feature or variable selection. Although the objective, to understand which are the influential variables, is often not the same as classification, the method has been successfully applied to that problem too.

*This chapter is prepared based on the materials presented in a *Workshop on Detecting Influential Variables in High-Dimensional Data* (held in June, 2009 at National Taiwan University, Taipei). We thank Hung Chen and Taida Institute of Mathematical Sciences for the support and organization of this very successful workshop. All authors of this chapter participated in the workshop.

S.-H. Lo (✉)
Department of Statistics, Columbia University, New York, USA
e-mail: slo@stat.columbia.edu

T. Zheng
Department of Statistics, Columbia University, New York, USA
e-mail: tzheng@stat.columbia.edu

H. Chernoff
Department of Statistics, Harvard University, Cambridge, Massachusetts, USA
e-mail: chernoff@stat.harvard.edu

I. Hu
Department of Information Systems, Business Statistics and Operations Management,
Hong Kong University of Science and Technology, Hong Kong
e-mail: imichu@ust.hk

I. Ionita-Laza
Department of Biostatistics, Columbia University, New York, USA
e-mail: ii2135@columbia.edu

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_5,
© Springer-Verlag Berlin Heidelberg 2011

5.1 Introduction

Advances in technology have led to an increasing availability in all scientific fields of high dimensional data, in which are buried important information. Developing effective methods of extracting hidden information from the vast amounts of messy and noisy data is a pressing challenge to statisticians.

In this chapter, we survey a general computer intensive approach, based on a method introduced in Lo and Zheng [17,18], for detecting which, of a large numbers of explanatory variables, have importance influence on a dependent variable Y. One feature of this approach consists of avoiding a difficult analysis involving hundreds or thousands of markers/explanatory variables in favor of a simple but effective analysis repeated many times. Another advantage is that, as opposed to other methods depending mainly on marginal signals from explanatory variables, this method detects and makes use of both marginal and interactive information to yield effective detections. For example, Lo and Zheng introduced a multi-locus method – the backward haplotype-transmission association (BHTA) algorithm – an efficient computationally intensive method of detecting important genes involved in complex human disorders [17, 18]. This method, using haplotype transmission information on multiple markers for affected subjects and their parents, was applied to an inflammatory bowel disease (IBD) data set. In the application of their 2004 paper, a total of 235 case-parent trios and 402 markers (variables) were included in the analysis. The findings of this application confirmed many previously identified IBD susceptibility loci reported in the literature and also included four novel loci (not previously reported) with exceptionally strong signals. It is worth noting that these four loci would not have been detected if only standard marginal methods were used. Furthermore, a recent genome-wide association study (GWAS), based on 3,230 cases and 4,829 controls, reported 30 SNP regions associated with the IBD [1].[1] It is interesting to note that, the four loci reported in Lo and Zheng [18] overlapped with four of the regions reported in Barrett et al. [1]. Given that the ratio of sample size and resolution in the two studies is large (the 2008 UK study is about 13-fold larger than that of the 2004 study), we believe that the BHTA has indicated the potential to extract hidden but valuable information by dint of considering interactions among markers, genes and variables.

The object of this chapter is to provide a comprehensive review on the development of a general framework with various data analyses, focusing on genetic applications and classifications. Section 5.2 provides a brief background on the subject of handling high-dimensional genetic data. Section 5.3 gives a review of the proposed general framework. Section 5.4 presents two real applications, focusing on the detection of gene-gene interactions and construction of association networks. We also introduce an important extension towards classification problems in Sect. 5.5. We conclude in Sect. 5.6 with a discussion of future directions and offer our views on current methodology issues in analyzing high dimensional genetic data.

[1] They actually used more subjects.

5.2 Background

5.2.1 Challenges from Current High-Dimensional Genetic Data

Technology developments in the past two decades have led to a rapid increase of data in genetics. Current genetic data are unprecedentedly high in dimension and are thus challenging to analyze. This is a trend likely to continue for decades to follow. We believe that current (and future) data contain vast amounts of information, and if analyzed and extracted effectively, would significantly improve our understanding of human life and causes of diseases. A number of common features are shared by these emerging genetic data. First, they are high dimensional – the number of variables and features can range from several hundreds to hundreds of thousands; Second, the sample sizes are usually quite small (in the tens) or moderate (several hundreds), and typically much smaller than the number of variables; Third, despite a large number of variables and features, most variables contribute noise rather than real signals, while only a small fraction of variables (1–2%, for instance) and their combinations are responsible for different outcomes of interest. Finally, for genetic data, the outcomes of certain activities are likely to be jointly determined by an unknown group of influential variables and features; these variables are correlated among themselves and thus require special attention in the statistical analysis. Consequently, methods that allow joint analysis incorporating interactive information among all variables become highly desirable.

5.2.2 Methods for Detecting Influential Variables

The most common practice today still remains the marginal analysis of the large number of variables in a genetic data set. New methods have emerged to address high dimensional data. In fact, most of the new methods addressing high dimensional data include a class of classification methods, with an aim for better prediction. Classification tasks have been concentrated primarily on searching for an accurate classifier rather than offering a better understanding of what and how the final classes are decided by groups of influential variables and their interactions [5, 7, 27]. We believe that such fundamental understanding is important, especially in genetic and medical applications where statistical and computational findings can usually provide further insights to reconfirm existing knowledge (such as interaction between proteins) or to form new hypotheses when compared with current beliefs.

In the following, we briefly summarize current classification (or related feature selection) methods and discuss their limitations for identifying influential genetic factors.

5.2.2.1 Ranking Genetic Factors Using Marginal Statistics

Most studies in biomedical research choose to use marginal procedures (such as t-test, correlation coefficient, χ^2 test of independence, etc) that evaluate the influence (importance) of a given single variable (factor) one at at time [6,7,12,26,27]. Information on joint influence or interactions between factors that is crucial for genetic reasoning is therefore ignored and lost. Not only do these methods suffer from a loss of efficiency due to their partial use of information, but they also offer little insight on the genetic contributions (and possible genetic models) of the genes identified.

5.2.2.2 Ranking Sets of Genetic Factors Based on Classification Accuracy

In most current classification driven genetic applications, much attention has been paid to evaluating prediction accuracy (e.g., [20]). It has also been proposed to rank a set of variables based on their ability to predict Y. The multifactor-dimensionality reduction (MDR) proposed in Ritchie et al. [23, 24] searches for susceptibility disease loci based on searching for a set of genetic factors that is most predictive of the disease outcome as evaluated by cross-validation. Such procedures address one of the needs in genetic/medical research, in terms of outcome prediction, but may fail to identify some of the important influential genetic factors on the outcome. This is because for a classification study on a sample of hundreds of observations, a good classifier does not necessarily depend on finding the variables that are most influential.

High dimensional machine learning methods such as support vector machines (SVM, e.g., [15, 30]), neural nets (NN, e.g., [13]) and random forests (e.g., [29]) have been used in genetic studies of high dimensional data [19]. They often show excellent prediction performance. However, it is usually hard to interpret the results in the genetic context. In particular for SVM and NN methods, the different nonlinear kernels used in the trained classifiers make it difficult to evaluate the results and generate testable genetic hypotheses. The method of random forests proposed in Breiman [2] has an embedded evaluation of each variable's importance. However, the algorithm is based on classification tree methods, which may not be easily associated with genetic interpretation.

5.2.3 The Development of Our Partition-Based Framework

Lo and Zheng introduced a multi-locus method for the case-parent trio data [17]. The case-parent trio design uses genetic information of an affected case and his/her parents to infer what genetic variants were transmitted to the case and what were not. By comparing the transmitted and untransmitted alleles, one can evaluate the association between genetic markers and the disease. To implement the multi-locus approach, they proposed to use a new information measure, Haplotype Transmission

Disequilibrium (HTD), for multiple markers based on the jointly transmitted and untransmitted genetic variants (i.e., *haplotypes*) and a screening algorithm, Backward Haplotye-Transmission Association (BHTA), based on HTD.

Family data such as that of the case-parent trio design are relatively harder and more expensive to collect, especially for diseases with late onsets. Recently, more attention has been focused on population-based case-control studies with a few thousand subjects in both groups. To accommodate such data, Zheng et al. proposed the backward genotype-trait association (BGTA) method, with a Genotype-Trait Distortion (GTD) score for capturing the information on the disease in multiple markers [31]. BGTA and its extensions have been applied to IBD [31], rheumatoid arthritis [4, 9, 21], breast cancer [16] and prostate cancer. In these studies influential genetic loci (represented by markers, usually) with significant interactions among them were detected. Chernoff et al. established the general theory and method framework based on the partitions and summarized all previously published methods as special cases [3].

With these methods, we can detect most influential variables and their interactions, and subsequently exposing the genetic and medical reasonings behind the classification of the symptoms. Our publications since 2002 represent a series of realistic statistical and computational efforts for a better understanding of genetic activities.

There are several major characteristics (in order of importance) of our approach. First, substantial portions of interactions among subgroups of lower dimensional spaces were captured and variables with weak signals were screened out. Second, our procedures consistently produce the information generated from influential variables and their interactions (or associations). An association network/genetic pathway as a standard output can then be easily constructed based on the aggregated information. Third, more efficient multi-stage procedures are studied.

5.3 Methodology Framework

In this section, we introduce a general framework of theory and methods. This is followed by a number of specialized methods with various applications.

We first introduce some notation. Let the potentially (candidate) influential variables be X_s, $1 \leq s \leq S$, which may have an influence on a dependent variable Y studied using a sample of n observations, $\mathbf{Z} = (\mathbf{X}, Y)$ where $\mathbf{X} = (X_1, X_2, \ldots, X_S)$. For clarity and simplicity, we shall use binary valued predicting/explanatory variables in the general discussion of this section. The extension to multilevel X variables is straightforward, as illustrated by the special case – the backward genotype-trait association (BGTA) method – introduced in Sect. 5.3.5, where the X's have three levels. It is commonly assumed that Y may be slightly or negligibly influenced by each of the individual variables of X_s, but may be profoundly influenced by the confluence of appropriate values within one or a few groups of these variables.

5.3.1 Evaluation of a Set of Variables' Influence on Y

Consider a subset of k binary valued variables from $X_1, X_2, \ldots, X_S$. For simplicity, we shall use $\{X_1, X_2, \ldots, X_k\}$ to denote this subset. These k selected variables define a partition Π^* of the sample of n observations into $m = 2^k$ subsets which we shall call partition elements $\{A_1, A_2, \ldots, A_m\}$ corresponding to the possible values of these k binary variables. Each partition element A_j corresponds to a possibly empty subset of n_j of the Y values and $\sum_{j=1}^{m} n_j = n$. Each nonempty partition element A_j yields a mean value $\bar{Y}_j$ and the overall mean $\bar{Y} = \sum_{j=1}^{m} n_j \bar{Y}_j / n$. The central *influence* measure of our approach is defined as

$$I_{\Pi^*} = \sum_{j=1}^{m} {n_j}^2 (\bar{Y}_j - \bar{Y})^2. \tag{5.1}$$

If I_{Π^*} is large, we suspect that some of the k variables may have an influence on Y.

Suppose that we now introduce another binary variable X_0, which leads to a more refined partition $\Pi = \{A_{jl} : 1 \le j \le 2^k, l = 0, 1\}$ where A_{j0} corresponds to that part of A_j where $X_0 = 0$ and A_{j1} corresponds to that part of A_j where $X_0 = 1$.

Now let n_{jl} be the number of elements in A_{jl} and $n_j = n_{j0} + n_{j1}$ be the number of elements in A_j. The measure I_{Π^*} is now replaced by

$$I_{\Pi} = \sum {n_{jl}}^2 (\bar{Y}_{jl} - \bar{Y})^2 \tag{5.2}$$

and

$$D_I = \frac{1}{2} (I_{\Pi} - I_{\Pi^*}) \tag{5.3}$$

can be regarded as a measure of how much X_0 contributes in influence on Y in the presence of $(X_1, X_2, \ldots, X_k)$. It is easy to see that

$$D_I = -\sum n_{j0} n_{j1} (\bar{Y}_{j1} - \bar{Y})(\bar{Y}_{j0} - \bar{Y}). \tag{5.4}$$

Thus, D_I tends to be negative when both means in the refined partition elements tend to be on the same side of $\bar{Y}$. We would expect that if the new variable contributes influence on Y, that D_I would tend to be positive.

Using the above properties of D_I, we can then start with an initial set of markers and iteratively eliminate variables that do not contribute to the influence on Y, one at a time. More specifically, if we commence with $k + 1$ variables, we consider the effect, i.e., D_I, of using the coarser partition obtained by eliminating one of the $k + 1$ variables. The variable with the smallest D_I is then eliminated; we repeat this procedure on the remaining k variables. We may repeat until we are satisfied by some criterion, (e.g., all the D_I are positive), that most of the remaining variables are good candidates for being influential (a local optimal subset). These remaining variables are *retained.*

Two cases are of special interest in genetic applications. One is that discussed earlier where the data consist of a sample of n independent observations. The other is the case where data are ascertained based on Y values. We consider here the results for n independent observations. For the latter case, the readers are referred to [3, Appendix A1].

First, we review and extend slightly our notation. The partition element A_{jk} yields n_{jk} observations with expectation $\mu_{jk} = E(Y|\mathbf{X} \in A_{jk})$, variance σ^2_{jk}, and sample mean $\bar{Y}_{jk}$. Then $n = \sum_{jk} n_{jk}$. Also,

$$\tilde{\mu} = n^{-1} \sum_{jk} (n_{jk}\mu_{jk}), \tag{5.5}$$

$$\tilde{\mu}_j = n_j^{-1}(n_{j0}\mu_{j0} + n_{j1}\mu_{j1}). \tag{5.6}$$

and $\tilde{\sigma}^2 = n^{-1} \sum_{jk} n_{jk}\sigma^2_{jk}$. We have used the tilde over the Greek letters μ and σ to emphasize those cases where these are conditional means and variances given the sample frequencies defined by $\mathbf{n} = \{n_{jk}\}$, but not parameters of the underlying distribution. We let the unconditional mean of Y be $\mu = E(Y)$.

Let

$$\varepsilon_j = \frac{n_{j1}}{n_j}(\mu_{j1} - \tilde{\mu}) - \frac{n_{j0}}{n_j}(\mu_{j0} - \tilde{\mu}).$$

A careful calculation yields

$$E(D_I|\mathbf{n}) = -H_1 + H_2 - n^{-1}G, \tag{5.7}$$

where

$$H_1 = \frac{1}{4}\sum n_j^2(\tilde{\mu}_j - \tilde{\mu})^2, \tag{5.8}$$

$$H_2 = \frac{1}{4}\sum n_j^2\varepsilon_j^2 \quad \text{and} \tag{5.9}$$

$$G = \sum n_{j0}n_{j1}(\tilde{\sigma}^2 - \sigma^2_{j0} - \sigma^2_{j1}). \tag{5.10}$$

Thus the conditional expectation of D_I, given the observable frequencies, involves three terms that consist of a heuristic decomposition explaining how the effects might change when variables are removed from current consideration. The first two terms are positive. The first, H_1, is related to the effect of $(X_1, ..., X_m)$ on Y in the presence of X_0. The second term, H_2, depends on the ε_j which are related to the effect of X_0 on Y in the presence of $(X_1, ..., X_m)$. The third term, $n^{-1}G$, is relatively small and can be estimated (and removed) from the data. In the extreme case where X_0 has no effect on Y in the presence of $(X_1, ..., X_m)$, the $n_j^2\varepsilon_j^2$ term equals $(n_{j1} - n_{j0})^2(\tilde{\mu}_j - \tilde{\mu})^2$, and the conditional expectation of D_I is negative (by neglecting G). The greater the effect of X_0 in the presence of $(X_1, ..., X_m)$, the more positive D_I tends to be. In the other extreme case where $\tilde{\mu}_j = \mu$ for all j, and hence $(X_1, ..., X_m)$ has no effect on Y, H_1 will be zero and the conditional expectation of D_I will be always positive, depending mainly on the marginal effect of X_0. Here the distribution of $\mathbf{X}$ did not play a major role. The results are relevant if the data arise from an experiment designed to select the explanatory variables in a systematic fashion.

5.3.2 Searching for Influential Variables in Random Subsets

In Sect. 5.3.1, we outlined the general statistic I used for evaluating a local subset of potentially influential variables. Often in reality, the number of variables and possible interactions in a large-scale study are much larger than that in a training set. The small/moderate number of observations causes a serious problem of sparseness when many variables are simultaneously evaluated. In these cases the real signals are diluted and swamped by the tremendous amount of noise generated by many uninformative variables. In order to measure the true signals due to influential variables and avoid the problems due to computational complexities, we must not consider too many variables at a time.

Within the learning scope posed by the sample size of the data, we concentrate on a small percentage of variables at a time. Our learning process will explore a large number of random subspaces (an idea similar to that of the random forests [2]). Within each random initialized subspace (a subset of variables), we search for the local cluster of variables that show high influence on the outcome Y, guided by D_I. It is likely that the resulting local optimal set is empty, indicating that no influential variable is found in this random subspace. Both the local cluster and its final I-score (denoted by I_{peak}) are recorded. Heuristically, the larger the I_{peak} score is the more influential of the identified local optimal cluster (of variables) is.

5.3.3 Resuscitation of Variables with Weak Marginal Signals

When the number of possible explanatory variables is very large, it may be computationally unfeasible to consider the interactions of pairs. Most methods reduce to considering only marginal effects to eliminate most of the available variables. In its original form the backward partition also can not expect to detect interactions if there are a limited number of influential variables. On the other hand if it is feasible to use some method, marginal or pairwise effects, to reduce the number of candidates, such a method may doom influential variables with little marginal effect. It is possible to resuscitate these “dead” variables by using the backward partition method where the random groups have some candidates drawn from the “live” variables and some from the “dead” ones.

5.3.4 Evaluation of Significance and Issue of Multiple Comparisons

In Chernoff et al. [3] we showed that the asymptotic distribution of I under the null hypothesis is a weighted sum of Chi-square random variables with one degree of freedom. This distribution can be derived using simulations or approximated by normal distributions when the number of partitions is large. For small data sets and large p values, this asymptotic distribution seems to work well. However, when the number of variables is large. the approximation of tail probabilities using the asymptotic distribution is problematic and leads to inaccurate p values for the most significant results. Therefore we recommend using permutations to evaluate the number of false positives due to chance and allow us to report estimated special form of false discovery rate (FDR) with our findings as we explained and demonstrated in our PNAS paper, [16].

Furthermore, when the joint effects of influential variables are weak, one cannot expect to reach statistical significance level after correction for multiple comparison in a single study. Instead, some clusters of variables with potential biological connections may be expected (with appropriate FDR, of course) and further replications studies are essential to confirm the findings (using independent datasets).

5.3.5 Special Case I: The Backward Genotype-Trait Association (BGTA) Method

We demonstrate in this section that the BGTA is indeed a special case of our general method.

Consider k biallelic markers under study. In a case-control genetic study, the genotype of each marker is used, which takes three values. Therefore, k markers define 3^k possible multi-marker genotypes. The BGTA algorithm is primarily based on a key statistic, the genotype-trait distortion (GTD) score based on the counts of these genotypes in the cases and controls:

$$\text{GTD} = \sum_{i=1}^{3^k} \left(\frac{n_{i,a}}{n_a} - \frac{n_{i,u}}{n_u} \right)^2, \tag{5.11}$$

where n_a and n_u are the number of cases and controls in the study, $n_{i,a}$ and $n_{i,u}$ are the counts of genotype i among the cases and controls respectively [31].

GTD measures the joint effect of a set of markers on the disease trait under study. It is easy to show that [3, Appendix S1]

$$I_\Pi = \sum_{j \in \Pi} n_j^2 \left(\bar{Y}_j - \bar{Y} \right)^2$$

$$= \sum_{i=1}^{3^k}(n_{i,a}+n_{i,u})^2\left(\frac{n_{i,a}}{n_{i,a}+n_{i,u}} - \frac{n_a}{n_a+n_u}\right)^2$$

$$= \frac{n_a^2 n_u^2}{(n_a+n_u)^2}\sum_{i=1}^{3^k}\left(\frac{n_{i,a}}{n_a} - \frac{n_{i,u}}{n_u}\right)^2, \qquad (5.12)$$

where Y is the affection status of the individuals, 1 if infected and 0 otherwise. Therefore, BGTA is a special case of the general partition-base framework.

Guided by the GTD score, the BGTA algorithm carries out a backward screening of the k markers and removes markers that lead to an increase to the GTD value. When the algorithm stops, it retains a *BGTA irreducible* set of markers (or cluster), removing any marker from which will contribute to a lower GTD score.

5.3.6 Special Case II: Linkage Analysis

The multilocus approach proposed by Lo and Zheng [17] has also been extended to a quite different disease mapping strategy, called *linkage analysis* [11]. Linkage analysis, similar to association analysis, is used to find regions in the genome that may harbor disease susceptibility loci. However, the goal in a linkage analysis is to identify broad regions linked to disease; follow-up association studies in those linked regions may identify particular variants associated with the trait.

A popular family design for linkage analysis is the *affected sib-pair* design using, for each family, two affected siblings and their parents. Linkage evidence is based on co-segregation of genetic materials from the parents to the affected siblings. The approach taken by Ionita and Lo [11] is a model-free linkage method, and the fundamental piece of information that we use is that of identity-by-descent (IBD) sharing. More precisely, two alleles, one from each of the sibs, are shared IBD, if they represent the same ancestral allele. Each parent transmits one allele to each of the sibs, and, under the null hypothesis of no linkage to disease, an IBD sharing of 0 or 1 is equally likely. For a marker, let n_0 be the total number of 0 IBD sharing in the dataset, and n_1 the corresponding number of 1 IBD sharing. This definition extends to multiple markers in a straightforward fashion and creates a partition as defined in our general framework.

As with the general approach by Lo and Zheng [17], the core of our method is the definition of a linkage measure that quantifies the linkage information contained in a set of markers. We define this multilocus measure as follows:

- For one marker, $H_1 = w_1(n_1 - n_0)^2$.
- For two markers, $H_{12} = w_2\left[\frac{(n_1^1-n_0^1)^2+(n_1^2-n_0^2)^2}{2} + (n_{11}^{12} - n_{00}^{12})^2\right]$.
- For k markers:

$$H_{1\ldots k} = w_k \left[\frac{\sum_{i=1}^{k} \left(n_1^i - n_0^i\right)^2}{\binom{k}{1}} + \frac{\sum_{i<j} \left(n_{11}^{ij} - n_{00}^{ij}\right)^2}{\binom{k}{2}} + \ldots \right.$$

$$\left. + \frac{\sum_{i_1<\cdots<i_{k-1}} \left(n_{1\ldots1}^{i_1\ldots i_{k-1}} - n_{0\ldots0}^{i_1\ldots i_{k-1}}\right)^2}{\binom{k}{k-1}} + \left(n_{1\ldots1}^{1\ldots k} - n_{0\ldots0}^{1\ldots k}\right)^2 \right]$$

where $n_{1\ldots1}^{i_1\ldots i_{k-1}}$ and $n_{0\ldots0}^{i_1\ldots i_{k-1}}$ are the numbers of $11,\ldots1$ and $00,\ldots,0$ IBD sharing at loci $i_1 \ldots i_{k-1}$, and w_i is a weight, $w_i = \frac{2^i}{2^i-1}$. This is equivalent to the average general influence measure I defined in (2) on the partitions based the multimarker identity-by-descent sharing status on all possible subsets of the k markers.

H defined above enjoys desirable properties similar to those of HTD in Lo and Zheng [17] and the general influence measure I defined in (2). It increases when noninformative markers are removed from the set and decreases when important markers are deleted. Ionita and Lo showed that this multilocus method may have increased power over traditional single-locus approaches, if the underlying disease model is indeed multilocus [11]. In an application to an Inflammatory Bowel Disease dataset, we have detected several of the susceptibility loci reported in the literature, in addition to a few novel regions.

5.4 Case Studies

In this section, we summarize two published studies on common human disorders using methods under the framework outlined in the previous section.

5.4.1 Inflammatory Bowel Disease

The inflammatory bowel disease (IBD) consists principally of ulcerative colitis (UC) and Crohn's disease (CD) – two chronic idiopathic inflammatory diseases of the gastrointestinal tract with overlapping features and shared complications. Epidemiological studies have shown that relatives of individuals with CD or UC are at increased risk for developing one or the other form of IBD, which suggests that at least some susceptibility genes will be shared by UC and CD. For a comprehensive review of IBD, readers are referred to [14, Chap. 15].

There have been multiple susceptibility loci with relevance to IBD etiology reported repeatedly in the literature since 1996, based on results from more than two dozen studies. These include IBD1 (16q12), IBD2 (12p13), IBD3 (6p21), IBD4

(14q11) and IBD7 (1p36). This suggested that the disease might be due to a number of genes with modest marginal effects.

In [18], Lo and Zheng applied the BHTA method [17] to a data set of 112 nuclear families with more than two Crohn's disease patients (89 with two patients, 20 with three patients and 3 with four patients), which is approximately 66% of the original dataset used in Rioux et al. [22] where linkage to IBD5 and IBD6 was reported. Among the patients, only those with parents on file can be used in BHTA algorithm, thus a total of 235 patient-parent trios were finally included in the analysis. Four hundred and two markers across the genome were evaluated in the screening.

The data had about 20% missing values. This is a more severe problem for multilocus methods like that of Lo and Zheng [17] than to single-marker methods. Conditioning on the observed genetic information within each family under study, the missing genetic information was imputed based on a random inheritance model under the null hypothesis. Multiple imputations were used to remove the effects of a single imputation. Screening results were the average of the screening outcomes on ten randomly fully imputed data sets.

Using BHTA, accounting for both joint and marginal effects, 48 (out of total 402) important markers that are potentially related to the disease susceptibility were identified. These 48 markers spread across many of the 23 Chromosomes (see Fig. 5.1) and overlap with all previously reported IBD loci, except IBD 6 (see Table 5.1), despite the fact that Rioux et al. [22] found no signal near IBD1, 2, 4 and 7. The discrepancy of the findings between these two studies provides convincing evidence in favor of the use of methods that take into account interactions and extract maximum amount of information available in the data.

Among the selected markers, the four markers that were most frequently returned, D1S549(1q32), D5S1470(5p15), D8S592(8q24), D21S1466(21q22), point to four novel loci, none of which have been reported in the present literature. Given that these signals were extremely strong, Lo and Zheng [18] suggested that "further research on these regions could be very fruitful." In [1], Barrett et al. used data from three studies on Crohn's disease and carried out an independent genomewide investigation using more than 500,000 SNP markers [1]. A total of 3,230 cases and 4,829 controls were used. They further used another independent data set of similar size to validate their results. They identified 30 significant SNP regions. The four novel regions (1q32.1, 5p15, 8q24, and 21q22) reported in Lo and Zheng [18] are among these regions with combined p-values 1.43×10^{-11}, 6.82×10^{-27}, 4.50×10^{-9} and 1.41×10^{-9}, respectively.

5.4.2 Breast Cancer

Breast cancer (MIM 114480) is another common and complex human disorder with several putative predisposing genes identified. It is generally believed that the risk of sporadic breast cancer is attributed to a complicated combined action of multiple genetic and environmental factors. From the National Cancer Institute, the Cancer

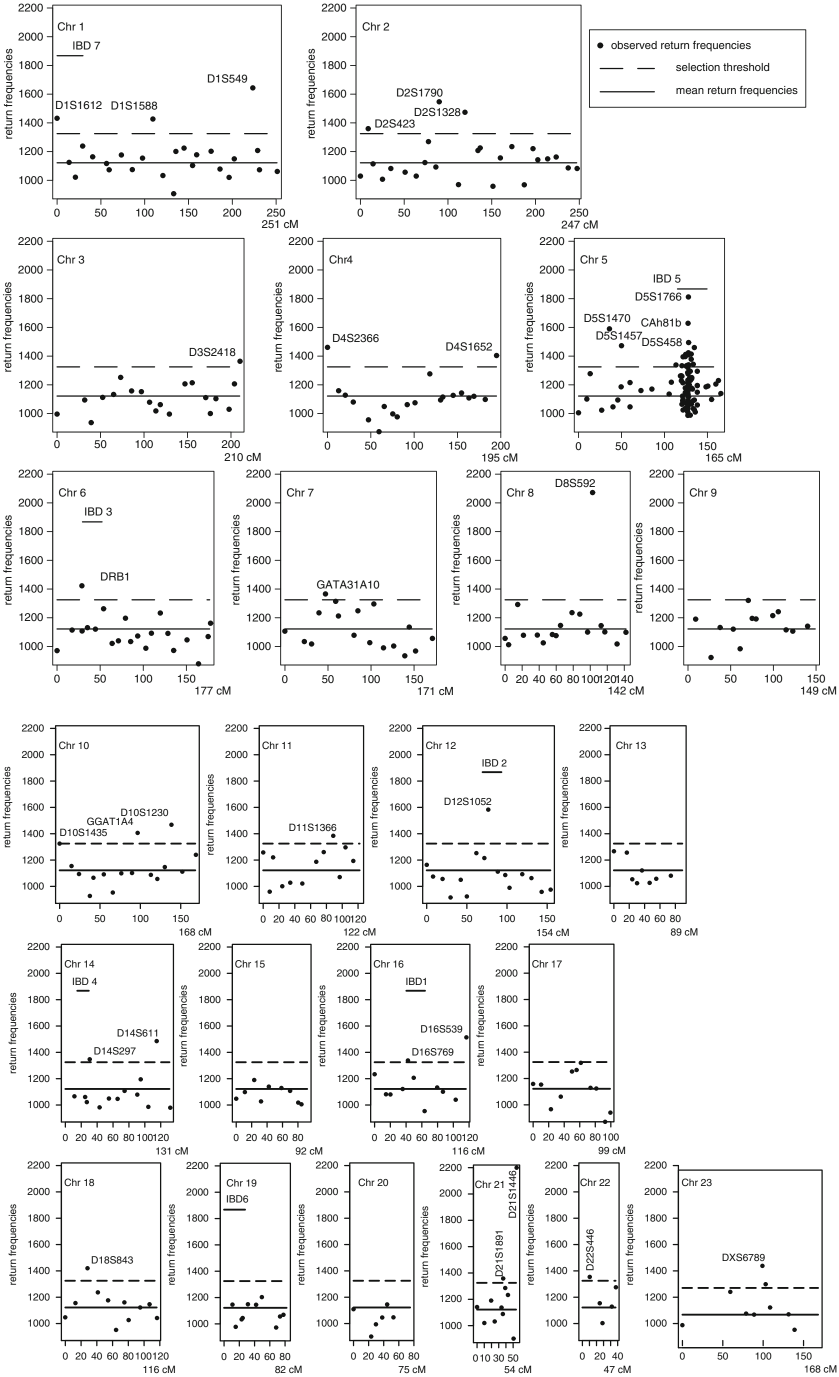


Fig. 5.1 IBD results (Reproduced with permission from Lo and Zheng [18])

Table 5.1 Returned markers on IBD loci

IBD locus	Selected marker
IBD 1 (16q12)	D16S769
IBD 2 (12p13)	D12S1052
IBD 3 (6p21)	DRB1
IBD 4 (14q11)	D14S297
IBD 5 (5q31)	CAh816[a]
IBD 6 (19p13)	
IBD 7 (1p36)	D1S1612

[a] 21 out 74 markers around IBD5 locus are selected

Genetic Markers of Susceptibility (CGEMS, http://cgems.cancer.gov/data/) initiative collected data on 1,145 cases of sporadic breast cancer and 1,142 controls and carried out a whole-genome association study using approximately 550,000 SNP markers [10]. In Lo et al. [16], a candidate gene study was carried out on selected SNPs from the CGEMS breast cancer data.

More specifically, 304 SNPs from 18 genes selected from the breast cancer literature were analyzed [16]. Since the SNPs markers are densely distributed on the genome, the genotypes of close-by SNPs are associated. Additionally, even though SNPs are useful in identifying mutations within genes, it is more geneticly relevant to study the interaction among genes as functional units. Therefore, Lo et al. aggregated statistics calculated on the SNP level into gene-based measures of both marginal effect and interaction effects [16]. First, GTD scores were calculated on each SNP individually and on each SNP pair. For each gene, its marginal effect would be the average GTD score of the SNPs that fall within the range of this given gene. Lo et al. observed that all SNPs and all SNP pairs have rather weak signals, indicating that none of these genetic loci or their pairwise interaction play a substantial role in deciding the risk of sporadic breast cancer [16]. Therefore, the significance of the excess signal from a SNP-SNP interaction were evaluated, conditioning on the SNPs' marginal signal. For each SNP pair, say (M_1, M_2), a new statistic was calculated:

$$r(M_1, M_2) = \frac{\mathrm{GTD}(M_1, M_2) - \mathrm{GTD}(M_1) \vee \mathrm{GTD}(M_2)}{\mathrm{GTD}(M_1) \vee \mathrm{GTD}(M_2)},$$

where "$\vee$" stands for maximum of the two values. This ratio measures the excess signal contained in two SNPs' interaction, compared to the strength of their marginal signals. The measure of gene-gene interaction is then the average of these SNP-pair level ratios.

Due to similar concerns of inter-SNP dependence as in Examples 2 and 3, the significance of the gene-gene interactions was evaluated using permutations and 16 significant pairs were found. These findings do not mean that these genes have strong joint effects on the risk of breast cancer. Rather, their interactions were found to have significantly more contribution to the risk of breast cancer than expected

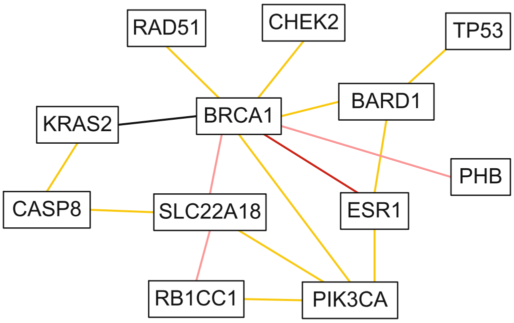


Fig. 5.2 Gene association network for a breast cancer candidate gene study (Example 3) (Reproduced from Lo et al. [16])

by chance, after controlled for their marginal signals. Based on these results, an association network was constructed as shown in Fig. 5.2.

For this study, instead of studying significant joint effects of genes (or SNP markers), gene-gene interaction with significant more effects on a disease's risk than the genes marginally were identified. This represents an alternative definition of "interaction" in such genetic epidemiology studies and provide a powerful method to find interactions that are significant but have moderate effects. Such interactions should be common for human disorders with complex disease etiology and may explain partly the limited success in finding disease-predisposing genes for such disorders.

5.5 Classification in Microarray Data Analysis: An Extension

This section contains a brief summary on how to apply the influence measure, HTD statistic, and the BHTA screening algorithm introduced in Lo and Zheng [17, 18] and Chernoff et al. [3] to classification problems. The details are given in Wang et al. [28]. The proposed classification method is intended to have two desirable properties. First, the classification rule derived from the method has a low error rate. Secondly, in the process of constructing the classification rule, influential variables responsible for the response are identified. That is, not only is the classification result accurate but also the classification rule contains important information for understanding the phenomenon under study.

5.5.1 Method Description

The method consists of four parts: feature selection, generating return sets from selected features, turning return sets into classifiers, and assembling classifiers to form the classification rule.

5.5.1.1 Feature Selection

Direct application of the BHTA to select influential variables may miss some key variable combinations when the number of variables is extremely large. We will calculate the influence score for each variable combination of a fixed size and select variables appearing frequently among the high-scoring combinations. Some details are given below.

It is necessary to first decide on the size of combinations. The larger the size, the higher the order of interactions we can detect, but the computation time increases by a factor depending on the total number of variables. For example, if we have 5,000 variables, then there are 12.5 million pairs and more than 20 billion triples. Although the influence scores of triples can provide information on third order interactions, the computational cost is more than 1,000 times that for pairs, which provide only information about second order interactions.

Simulated and real data sets have suggested that once a group size is chosen, The peak I values after reduction for a large sample of groups have a distribution where the cumulative grows rapidly until some relatively large values are reached. Those large values suggest a threshold, below which we should neglect the groups. We then look at the retention frequencies for the retained variables in the satisfactory groups, and rate these variables by how often they are retained in these groups. The cumulative of these frequencies also tends to yield a similar threshold, indicating a reduced set of variables worth analyzing.

5.5.1.2 Generate Return Sets

We now apply the BHTA algorithm to the selected high-frequency variables. There are two parameters to be determined before BHTA can be applied: the starting size and the number of repetitions. The starting size refers to how many variables we select from the high-frequency-variable pool so that BHTA algorithm can be applied. The starting size depends on the number of training cases. If the starting size is too large, then most cells (or partition elements) in the partition by the set of variables after dropping one, contain at most one training case. In this case, the HTD scores before and after dropping one variable differ very little. Therefore, the algorithm basically does random dropping in the first few steps and there is a substantial chance of dropping an influential variable and thus of missing a key variable-combination. The ideal starting size is such that one can expect several cells with two or more cases after dropping one variable.

The second quantity to be determined is the number of repetitions for the BHTA, which depends on the number of training cases also. The number of training cases determines the maximum size of return sets that can be supported. For example, if we have 100 training cases, then they can support a return set of size 4, when each explanatory variable is binary. Each return set of size 4 has $2^4 = 16$ cells. Thus, on the average, each cell contains more than five training cases. The HTD statistic is reliable following the rule of thumb that the chi-square statistic is dependable, if on the average each cell contains five or more observations. Hence the return set of size 4 is well supported by the training set of 100 cases and we want to make sure that the BHTA algorithm is repeated sufficient number of times so that combinations of size 4 are covered rather completely.

After determining the starting size and the number of repetitions, we can run the BHTA algorithm. The return sets generated from BHTA will undergo two filtering procedures to eliminate inferior ones. For details on filtering procedures and the determination of staring sizes and the number of repetitions, please see [28].

5.5.1.3 Turning Return Sets into Classifiers

There are several ways to construct classifiers from return sets. A classification tree classifies a test case using only information from the cell containing the test case. It classifies a case based on a few training cases and cannot "borrow strength" from other cells. When explanatory variables are continuous, we lose information from the tree classifier when we discretize variables. However, if all variables are discrete, it seems better to use tree classifiers.

The SVM is sensitive to tuning parameter selection. It usually was not as effective as logistic regression in datasets that we tested. Given the popularity of SVM and its elegant theoretical properties, this is quite surprising. In the preliminary study, the logistic regression sometimes produced unreliable results when there was a perfect separating hyperplane or when the AIC was several times larger than the usual values. Linear discriminant analysis (LDA) is numerically more stable, but the training-set error rate is usually larger than for logistic regression in the microarray datasets we studied.

The classifier selection situation we faced here is quite different from that elsewhere. The characteristics of the situation are (a) there may exist high-order interactions among variables without lower-order interactions; (b) the classifier needs to complement well classifiers from other return sets. Our classifier selection strategy is formulated with this unique perspective in mind.

5.5.1.4 Assemble Classifiers to Form the Classification Rule

The previous section describes how to construct classifiers from return sets. We now describe how to combine classifiers to form a classification rule. In machine learning, the method that fits our objective best is termed boosting; see, e.g., [8]. The AdaBoost minimizes an exponential loss function to find a sequence of basis functions along with optimal weights to form an additive representation of the classification rule. The basis functions here are classifiers obtained from return sets. For details on the boosting algorithm under the current setting, please see [28].

Sometimes, the classifier constructed from a return set has zero error rate in the training set. The weight for such a "perfect" classifier is not well-defined according to the boosting method. We resolved the issue in a heuristic manner: taking the number of wrongs to be 1/2 instead of 0.

5.5.2 Real Data Examples

This section contains three real-data examples. The first two are microarray gene expression datasets for breast cancer. They are from van ’t Veer et al. [27] and Sotiriou et al. [25], which originally contain 24,187 genes and 97 patients, and 7,650 genes and 99 patients, respectively. The purpose for the van ’t Veer et al. [27] is to classify female breast cancer patients into relapse and non-relapse types using gene expression data, while that of Sotiriou et al. [25] is to classify tumor subtypes.

In the van ’t Veer et al. dataset, after initial screening, 4,917 genes were kept for the classification task [27]. Following van ’t Veer et al. [27], 78 cases out of 97 are used as the training sample (34 relapse, 44 non-relapse) and 19 (12 relapse and 7 non-relapse) as the test sample. Our method yields 10.5% error on the test sample, which corresponds to the best error rate reported in the literature. To test the stability of our method, we further randomly selected 10 test samples of size 10 each. The remaining $97 - 10 = 87$ cases were used as the training samples. That is, we randomly split the data set into two groups of 10 and 87 cases, respectively. Construct the classification rule using the proposed method on the group of 87 cases and then test the classification rule on the other ten cases. The whole procedure is repeated ten times and the average error rate based on the ten test samples is 10%.

The Sotiriou et al. dataset contains 7,650 genes on 99 patients [25]. The task is to classify tumors according to their estrogen receptor (ER) status using gene expression information. This is different from the objective of van ’t Veer et al. [27], where the goal is to discriminate relapse patients from non-relapse ones. We randomly selected 10 patients as the test set and keep the remaining 89 as the training set. The error rate is around 7% based on six test sets of size 10 each.

The third dataset is from [7], which consists of 7,129 genes, 38 samples in the training set and 34 samples in the test set. The purpose is to classify acute leukemia into two subtypes: acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Our classification rule missclassifies one test case, which matches the best result in the literature.

In all three examples, the classification rule is constructed using only the information from the training sample and no information whatsoever from the test sets; see [28] for details.

5.5.3 Summary of the Classification Method

The proposed classification method has several distinguishing features. First, the error rates in several real-data examples are at least as good as the best results in the literature or even better. Secondly, the variables included in the classification rule are selected using the influence measure I, which is effective in identifying influential variables. Thirdly, the error rate estimates are well calibrated and free of selection bias.

5.6 Conclusion

Most of the methods in use today for determining which of many possible explanatory variables influence a dependent variable, are based on the marginal effect of each candidate variable. These methods fail to take into account the possibility that an influential variable has little marginal effect, but can be very effective when interacting with another variable.

There is a small class of methods that try to take these interactions into account. They tend to be of limited use when the number of candidate variables is very large and most of the influential variables have little marginal effect.

The backward partition method and its variants have proven to be very effective in producing results in several important biological problems. Although this method is not primarily directed toward the classification problem, the results of this method when applied toward classification have given results at least as good as those published.

Acknowledgements The research is partially supported by NSF grant DMS 0714669, NIH Grant R01 GM070789 and ARRA supplement 3R01GM070789-05S1, and Hong Kong RGC grant 642207. We thank our graduate students, Chien Hsun Huang and Haitian Wang (Maggie) for their diligence in our research activities that have led to several publications discussed in this chapter and their help during the workshop.

References

1. Barrett, J. C., Hansoul, S., Nicolae, D. L., Cho, J. H., Duerr, R. H., Rioux, J. D., Brant, S. R., Silverberg, M. S., Taylor, K. D., Barmada, M. M., Bitton, A., Dassopoulos, T., Datta, L. W., Green, T., Griffiths, A. M., Kistner, E. O., Murtha, M. T., Regueiro, M. D., Rotter, J. I., Schumm, L. P., Steinhart, A. H., Targan, S. R., Xavier, R. J., Libioulle, C., Sandor, C., Lathrop, M., Belaiche, J., Dewit, O., Gut, I., Heath, S., Laukens, D., Mni, M., Rutgeerts, P., Van Gossum, A., Zelenika, D., Franchimont, D., Hugot, J. P., de Vos, M., Vermeire, S., Louis, E., Cardon, L. R., Anderson, C. A., Drummond, H., Nimmo, E., Ahmad, T., Prescott, N. J., Onnie, C. M., Fisher, S. A., Marchini, J., Ghori, J., Bumpstead, S., Gwilliam, R., Tremelling, M., Deloukas, P., Mansfield, J., Jewell, D., Satsangi, J., Mathew, C. G., Parkes, M., Georges, M., & Daly, M. J. (2008). Genome-wide association defines more than 30 distinct susceptibility loci for Crohn's disease. *Nature Genetics*, *40*(8), 955–962.
2. Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.
3. Chernoff, H., Lo, S. H., & Zheng, T. (2009). Discovering influential variables: A method of partitions. *Annals of Applied Statistics*, *3*(4), 1335–1369.

4. Ding, Y., Cong, L., Ionita-Laza, I., Lo, S. H., & Zheng, T. (2007). Constructing gene association networks for rheumatoid arthritis using the backward genotype-trait association (BGTA) algorithm. *BMC Proceedings*, *1*(Suppl 1), S13.
5. Dudoit, S., Fridlyand, J., & Speed, T. P. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, *97*(457), 77–87.
6. Efron, B., & Tibshirani, R. (2002). Empirical bayes methods and false discovery rates for microarrays. *Genetic Epidemiology*, *23*(1), 70–86.
7. Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., Bloomfield, C. D., & Lander, E. S. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, *286*(5439), 531–537.
8. Hastie, T., Tibshirani, R., & Friedman, J. H. (2003). *The elements of statistical learning* (corrected ed.) New York, NY: Springer.
9. Huang, C. H., Cong, L., Xie, J., Qiao, B., Lo, S. H., & Zheng, T. (2009). Rheumatoid arthritis-associated gene-gene interaction network for rheumatoid arthritis candidate genes. In *BMC proceedings for the genetic analysis workshop 16, Vol.*. BMC Proceedings 2009, *3*(Suppl 7):S76 (15 December 2009)
10. Hunter, D. J., Kraft, P., Jacobs, K. B., Cox, D. G., Yeager, M., Hankinson, S. E., Wacholder, S., Wang, Z., Welch, R., Hutchinson, A., Wang, J., Yu, K., Chatterjee, N., Orr, N., Willett, W. C., Colditz, G. A., Ziegler, R. G., Berg, C. D., Buys, S. S., McCarty, C. A., Feigelson, H. S., Calle, E. E., Thun, M. J., Hayes, R. B., Tucker, M., Gerhard, D. S., Fraumeni, J. F., Jr., Hoover, R. N., Thomas, G., & Chanock, S. J. (2007). A genome-wide association study identifies alleles in FGFR2 associated with risk of sporadic postmenopausal breast cancer. *Nature Genetics*, *39*(7), 870–874.
11. Ionita, I., & Lo, S. H. (2005). Multilocus linkage analysis of affected sib pairs. *Human Heredity*, *60*(4), 227–240.
12. Kerr, M. K., & Churchill, G. A. (2001). Statistical design and the analysis of gene expression microarray data. *Genetical Research*, *77*(2), 123–128.
13. Khan, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C. R., Peterson, C., & Meltzer, P. S. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, *7*(6), 673–679.
14. King, R. A., Rotter, J. I., & Motulsky, A. G. (2002). *The genetic basis of common diseases* (2nd ed.). New York, NY: Oxford University Press.
15. Lee, Y., & Lee, C. K. (2003). Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Bioinformatics*, *19*(9), 1132–1139.
16. Lo, S. H., Chernoff, H., Cong, L., Ding, Y., & Zheng, T. (2008). Discovering interactions among BRCA1 and other candidate genes associated with sporadic breast cancer. *Proceedings of the National Academy of Science United States of America*, *105*(34), 12,387–12,392.
17. Lo, S. H., & Zheng, T. (2002). Backward haplotype transmission association (BHTA) algorithm – a fast multiple-marker screening method. *Human Heredity*, *53*(4), 197–215.
18. Lo, S. H., & Zheng, T. (2004). A demonstration and findings of a statistical approach through reanalysis of inflammatory bowel disease data. *Proceedings of the National Academy of Science United States of America*, *101*(28), 10,386–10,391.
19. McKinney, B. A., Reif, D. M., Ritchie, M. D., & Moore, J. H. (2006). Machine learning for detecting gene-gene interactions: A review. *Applied Bioinformatics*, *5*(2), 77–88.
20. Pochet, N., De Smet, F., Suykens, J. A. K., & De Moor, B. L. R. (2004). Systematic benchmarking of microarray data classification: Assessing the role of non-linearity and dimensionality reduction. *Bioinformatics*, *20*(17), 3185–3195.
21. Qiao, B., Huang, C. H., Cong, L., Xie, J., Lo, S. H., & Zheng, T. (2009). Genome-wide gene-based analysis of rheumatoid arthritis-associated interaction with PTPN22 and HLA-DRB. In *BMC proceedings for the genetic workshop analysis 16, Vol.*. BMC Proceedings 2009, *3*(Suppl 7): S132.

22. Rioux, J. D., Silverberg, M. S., Daly, M. J., Steinhart, A. H., McLeod, R. S., Griffiths, A. M., Green, T., Brettin, T. S., Stone, V., Bull, S. B., Bitton, A., Williams, C. N., Greenberg, G. R., Cohen, Z., Lander, E. S., Hudson, T. J., & Siminovitch, K. A. (2000). Genomewide search in canadian families with inflammatory bowel disease reveals two novel susceptibility loci. *American Journal of Human Genetics*, *66*(6), 1863–1870.
23. Ritchie, M. D., Hahn, L. W., & Moore, J. H. (2003). Power of multifactor dimensionality reduction for detecting gene-gene interactions in the presence of genotyping error, missing data, phenocopy, and genetic heterogeneity. *Genetic Epidemiology*, *24*(2), 150–157.
24. Ritchie, M. D., Hahn, L. W., Roodi, N., Bailey, L. R., Dupont, W. D., Parl, F. F., & Moore, J. H. (2001). Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. *American Journal of Human Genetics*, *69*(1), 138–147.
25. Sotiriou, C., Neo, S. Y., McShane, L. M., Korn, E. L., Long, P. M., Jazaeri, A., Martiat, P., Fox, S. B., Harris, A. L., & Liu, E. T. (2003). Breast cancer classification and prognosis based on gene expression profiles from a population-based study. *Proceedings of the National Academy of Science United States of America*, *100*(18), 10,393–10,398.
26. Tusher, V. G., Tibshirani, R., & Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(9), 5116–5121.
27. van 't Veer, L. J., Dai, H. Y., van de Vijver, M. J., He, Y. D. D., Hart, A. A. M., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., & Friend, S. H. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, *415*(6871), 530–536.
28. Wang, H., Lo, S. H., Zheng, T., & Hu, I. (2009). *A new classification method incorporating interactions among variables for high-dimensional data*. Working paper.
29. Zhang, H., Yu, C. Y., & Singer, B. (2003). Cell and tumor classification using gene expression data: Construction of forests. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(7), 4168–4172.
30. Zhang, H. H., Ahn, J., Lin, X., & Park, C. (2006). Gene selection using support vector machines with non-convex penalty. *Bioinformatics*, *22*(1), 88–95.
31. Zheng, T., Wang, H., & Lo, S. H. (2006). Backward genotype-trait association (BGTA)-based dissection of complex traits in case-control designs. *Human Heredity*, *62*(4), 196–212.

Chapter 6
STORMSeq: A Method for Ranking Regulatory Sequences by Integrating Experimental Datasets with Diverse Computational Predictions

Jim C. Huang and Brendan J. Frey

Abstract We present STORMSeq (STructured ranking of Regulatory Motifs and Sequences), a novel probabilistic method for sequence search in which we learn to rank sequences using heterogeneous experimental datasets and the outputs of diverse computational prediction methods. By formulating the problem of sequence search as one of ranking, STORMSeq largely avoids issues of model misspecification and complex inference which arise when modelling different types of datasets in the presence of many hidden variables. The framework allows one to compare orderings over sequences conveyed by diverse types of data, though the data measurements and scoring systems may be difficult to compare to one another. We demonstrate STORMSeq in the contexts of scoring sequences bound by transcription factors and for the problem of finding microRNA targets in human retinoblastomas where in the latter case we can combine mRNA and microRNA expression with protein abundances and sequence data. We will show for both of these problems that (a) by accounting for the dependencies inherent in learning to rank and (b) by incorporating multiple datasets with computational predictions, we can improve the accuracy with which we rank sequences compared to standard methods. Our method is general and can be applied to a wide variety of other problems in which heterogeneous data sets are available, such as ranking therapeutic drug targets and discovery of genetic associations to disease.

B.J. Frey (✉)
Probabilistic and Statistical Inference Group, University of Toronto, 10 King's College Road, Toronto, ON, M5S 3G4, Canada
e-mail: frey@psi.toronto.edu

J.C. Huang
Microsoft Research, One Microsoft Way, Redmond, WA, 98052, USA
e-mail: jimhua@microsoft.com

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_6,

© Springer-Verlag Berlin Heidelberg 2011

6.1 Introduction

The problem of sequence search, such as discovering transcription factor (TF) binding sites, microRNA targets and structural genetic variants, remains a significant challenge in genomics. Several *de novo* computational methods have been developed with the aim of searching for overrepresented sequences using sequence data [2, 12]. Due to the degeneracy of such sequences, such methods often require the use of sequence conservation in order to minimize false positive rates. To address this, computational methods have recently begun to account for additional features such as the accessibility of target sequences due to RNA secondary structure [14], contextual features [7] or other types of quantitative profiling data [5, 9, 19, 20]. As newer methods for discovering sequences and new profiling technologies continue to emerge, the issue of how to update existing sequence search methods to account for multiple types of data remains a significant challenge. In addition to accounting for several types of data, incorporating the large number of computational predictions already available will also be desirable.

6.1.1 *Previous Work*

In recent years, many different methods have been proposed to address the problem of integrating together large heterogeneous datasets in the context of sequence search. For example, probabilistic generative models have been proposed in which sequence search consists of inference and learning [9, 19] given sequence and expression data. Although such methods explicitly model the impact of sequences on gene expression while accounting for uncertainty, a major challenge is to account for newer datasets as well as new sources of regulatory variability. Each additional dataset to be analyzed is likely to introduce a significant number of additional parameters and hidden variables, dramatically increasing the cost and complexity of inference and learning under the generative framework. Thus, as the number of types and sizes of data continue to increase, it is likely that both model misspecification and prohibitive computational complexity will hamper the practicality of probabilistic models with latent variables for discovering sequences. Owing to the difficulty in developing purely sequence-based models of regulatory sequences, a major challenge is then to incorporate additional data types under a unified tractable and principled framework.

6.1.2 *Sequence Search as a Problem of Learning to Rank*

A strategic approach to the above problem can be obtained by noting that the problem of sequence search is inherently a problem of learning to rank, whereby we are given a large number of possible sequences and only some relatively small number are of biological significance. Furthermore, there is often a well-defined notion of preference between sequences. An example of this arises when searching for transcription factor sites, whereby some sites are more strongly bound than others by certain transcription factors. Thus when discovering sequences, it is desirable to explicitly model the fact that sequences do not fall into two distinct categories of positives and negatives but instead have different degrees of significance attached to them, so that a plausible model should assign a higher score for sequences with higher importance.

Some methods have in fact formulated the problem of sequence search as one of ranking, so that they assign a score to each sequence with the implicit assumption that high-scoring sequences are more likely to be *bona fide* than low-scoring ones. The idea of discovering sequences using an explicit ranking formulation has been explored previously by [3,5,20] in the context of using the orderings obtained from microarray intensities to learn position-specific scoring matrices (PSSMs) for transcription factor binding sites (TFBS). This was shown to significantly improve predictive accuracy with respect to other model-based methods, as no assumptions on the functional relationship between measured intensities and sequences needed to be made in order to learn to rank sequences. The improved accuracy of such ranking-based methods with respect to model-based methods then suggests that a good method for discovering sequences would be one specifically tailored to the problem of learning to rank.

Given the above methods for ranking sequences, our goal here is to expand on previous work along three directions. First, we address the presence of statistical dependence relationships between variables in the problem of ranking, since the rank of one sequence can only be determined given the ranks of all sequences. Second, the scoring function used by the previous methods of [3,5,20] was parameterized by a PSSM and so only accounted for sequence inputs. Here we will allow for ranking functions which can account for rich feature spaces obtained from quantitative measurements such as expression profiling data. Lastly, by formulating the sequence search problem as one of ranking, we can leverage information across several experimental datasets and diverse prediction methods via the orderings over sequences that each provides. Under the framework of learning to rank, orderings provided by diverse computational methods and those provided by experimental data are all comparable and readily accounted for, even if measured/predicted values between different prediction methods may be difficult to compare. Thus, given that we observe many different partial orderings provided by diverse datasets and prediction methods, our aim will be to predict orderings over sequences so that sequences which are often highly ranked across different experiments and prediction methods should also be highly ranked by our method. The proposed framework of ranking then offers three significant advantages over previous approaches for sequence search. First, the framework makes minimal assumptions about the relationships between sequences and measured/predicted labels for the sequences and so largely avoids the issue of model misspecification. Second, it allows us to leverage orderings provided by heterogeneous datasets and prediction methods which may have little overlap with one another in the sequences they contain, but are nevertheless informative when combined together under a single model. Lastly, predictive accuracy is improved by explicitly modelling the dependencies involved in learning to rank.

To model the statistical dependencies in learning to rank, we can take advantage of the structured ranking learning framework which was recently proposed in [11]. This probabilistic framework for learning to rank is based on a novel class of probabilistic graphical models called cumulative distribution networks [8, 10], or CDNs. In learning to rank in a structured setting where we account for dependence relationships between model variables, or *structured ranking learning*, the goal is to learn a ranking function under a structured loss functional which accounts for the statistical dependence relationships involved in predicting pairwise preferences between sequences, as misranking one sequence affects how we rank other sequences. In the context of discovering sequences, we can then interpret a set of prediction methods and a set of experimental measurements as observations which convey partial orderings over some subset of the sequences of interest. Thus we present STORMSeq, a method formulated which scores sequences given a set of features and a set of orderings over subsets of the sequences to be ranked. Our method generalizes the RankMotif++ method of [5] to a structured learning setting where we can (a) account for the dependencies in the problem of ranking, (b) incorporate rich feature spaces such as quantitative measurements of mRNA and protein expression in addition to sequence data, and (c) account for diverse computational prediction methods as additional data. The outline of the method is illustrated in Fig. 6.1. We will apply the proposed framework to the problems of scoring transcription factor binding sites and microRNA targets, although the framework is general enough to be applied to a wide variety of bioinformatics problems, such as ranking therapeutic drug targets, finding genetic associations or scoring protein-protein interactions.

6.2 STORMSeq: STructured Ranking of Regulatory Motifs and Sequences

We will begin by describing the problem of structured ranking learning for discovering sequences using the framework of [11]. Suppose we wish to score sequences in the set $\mathscr{S}$. Let s_α be a particular sequence in $\mathscr{S}$ which is indexed by α. Here, a sequence is any segment of nucleotides or amino acids for which one can extract features. For example, in the case where we wish to discover microRNA targets, a ‘sequence’ may correspond to the entire $3'$ untranslated region ($3'$UTR) for a particular gene, so that one has access to the sequence of the $3'$UTR, as well as other features for the $3'$UTR sequence. These can include its level of expression across many tissues/cell types, the abundance of proteins which are translated from the sequence preceding the $3'$UTR and the expression of a microRNA which putatively targets a site in the $3'$UTR sequence. This is illustrated in Fig. 6.2a: for each node α, we are provided with a corresponding sequence s_α and a set of features $\mathbf{x}_\alpha$ which will aid in learning to rank the sequences.

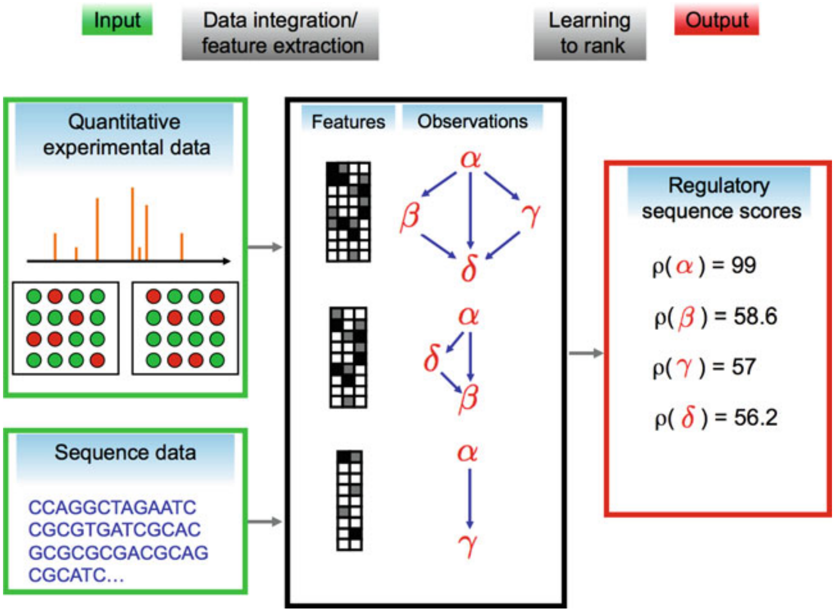


Fig. 6.1 The STructured ranking of Regulatory Motifs and Sequences (STORMSeq) method. Given multiple independent observations conveying various orderings over sequences and given the observed sequences and input features extracted for each observation (e.g.: mRNA, microRNA and protein measurements, sequence context features), STORMSeq learns a ranking function such that the probability of generating the observed orderings is maximized

Suppose now that we are given a set of N observations $\mathscr{D} = \{D_1, \ldots, D_N\}$, where each observation D_n provides an ordering of the sequences in some subset $\mathscr{S}_n \subseteq \mathscr{S}$. Here, an observation contains a partial ordering of the sequences to be ranked. For example, in the context of scoring microRNA targets, orderings might be provided by gene expression values in microRNA overexpression experiments [9] or they can be provided by scores output by computational prediction methods [7, 16, 18]. The orderings over sequences in an observation can then be viewed as a set of pairwise preference relationships between sequences, which we will denote using $\alpha \succ \beta$. For a given observation, we can then represent the ordering between sequences as a directed graph in which a directed edge $e = (\alpha \rightarrow \beta)$ is drawn between two nodes α, β if sequence s_α was *preferred* to sequence s_β within observation D_n. We will denote this directed graph as the order graph $G_n = (V_n, E_n)$ for observation D_n, where E_n is the set of all edges in the order graph and each node $\alpha \in V_n$ corresponds to a unique sequence $s_\alpha \in \mathscr{S}_n$. An example of such an order graph is shown in Fig. 6.2b. Thus the nth observation consists of the set $D_n = \{G_n, \{s_\alpha, \mathbf{x}_\alpha\}_{\alpha \in V_n}\}$, so that our data consists of a collection of independent observations $\mathscr{D} = \{D_1, \ldots, D_N\}$. One immediate advantage of the proposed framework is that orderings over sequences can be compared between observations despite the fact that measured/predicted values between observations may not be

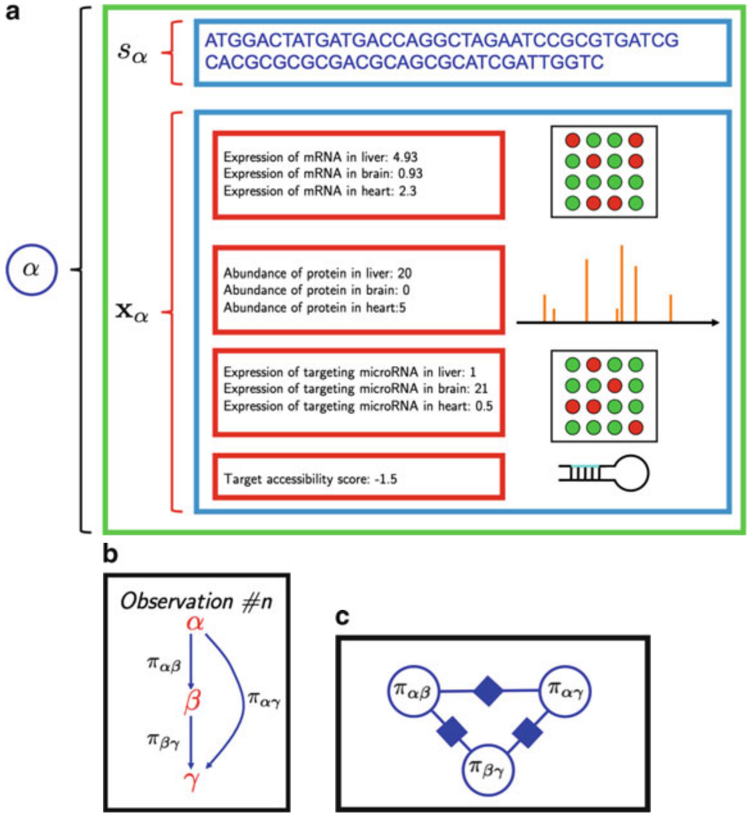

a
s_α
ATGGACTATGATGACCAGGCTAGAATCCGCGTGATCG
CACGCGCGCGACGCAGCGCATCGATTGGTC
α
$\mathbf{x}_\alpha$
Expression of mRNA in liver: 4.93
Expression of mRNA in brain: 0.93
Expression of mRNA in heart: 2.3
Abundance of protein in liver: 20
Abundance of protein in brain: 0
Abundance of protein in heart:5
Expression of targeting microRNA in liver: 1
Expression of targeting microRNA in brain: 21
Expression of targeting microRNA in heart: 0.5
Target accessibility score: -1.5
b
Observation #n
α
$\pi_{\alpha\beta}$
β
$\pi_{\alpha\gamma}$
$\pi_{\beta\gamma}$
γ
c
$\pi_{\alpha\beta}$
$\pi_{\alpha\gamma}$
$\pi_{\beta\gamma}$

Fig. 6.2 The STructured ranking of Regulatory Motifs and Sequences (STORMSeq) method. (**a**) Feature extraction. For each sequence s_α to be ranked, we assign a corresponding node α and a set of corresponding features which are relevant to ranking the sequence. For the example shown, the sequence s_α may correspond to the sequence for the entire 3′ untranslated region (3′UTR) of a gene, so that the feature vector $\mathbf{x}_\alpha$ include the expression of the gene carrying the sequence, the abundance of protein produced from the coding region for the gene carrying the sequence and the expression of a putative microRNA which targets the sequence; (**b**) An observation consisting of an order graph over three nodes where each node α, β, γ in the order graph corresponds to a unique sequence $s_\alpha, s_\beta, s_\gamma$ to be ranked, and each directed edge expresses a preference relationship between two nodes. An order graph can be readily established from log p-value scores, expression ratios or other available statistics which provide an indication of the relevance or importance of a given sequence. In this example the order graph corresponds to the ordering $\alpha \succ \beta \succ \gamma$. Each edge in the order graph then corresponds to preference variables $\pi_{\alpha\beta}, \pi_{\beta\gamma}, \pi_{\alpha\gamma}$; (**c**) The corresponding cumulative distribution network (CDN) defined over the preference variables specified by the observation of (**b**). The CDN models the joint CDF over the preference variables and allows us to compactly specify dependencies between preferences so we can perform structured ranking learning [11]

comparable. Furthermore, the orderings conveyed by different observations can be partial and can be defined over different subsets of sequences.

To combine the different orderings together, we now define a *ranking function* $\rho(\alpha) : V_n \rightarrow \mathbf{R}$ which assigns real-valued scores to sequences. If we model the stochastic score σ_α of a given node α as

$$\sigma_\alpha = \rho(\alpha) + \pi_\alpha, \tag{6.1}$$

where π_α is a random variable specific to node α, then we can define the preference event $\alpha \succ \beta$ as being equivalent to the following:

$$\alpha \succ \beta \Leftrightarrow \pi_{\alpha\beta} \equiv \pi_\beta - \pi_\alpha \leq \rho(\alpha) - \rho(\beta). \tag{6.2}$$

Here, $\pi_{\alpha\beta}$ is a *preference variable* between α, β. Thus for each edge (α, β) in the order graph G_n, we assign a corresponding continuous-valued preference variable $\pi_{\alpha\beta}$ which should satisfy the above inequality in order for the preference relation $\alpha \succ \beta$ to be observed. Now we can define the quantity $r(e; \rho, D_n) = \rho(\alpha) - \rho(\beta)$ and collect these into a vector $\boldsymbol{r} \equiv \boldsymbol{r}(D_n; \rho) \in \mathbf{R}^{|E_n|}$ of pairwise differences, where $|E_n|$ is the number of edges in the order graph. Similarly, let $\pi_e \equiv \pi_{\alpha\beta}$ be the preference variable defined along edge e in the order graph G_n. Having defined the preference variables, we must now select an appropriate loss measure for learning the ranking function. For a given observation D_n, we will choose the loss measure to be the negative log-probability of observing the preference relationships between sequences in order graph G_n. From Eq. 6.2, this will take the form of a probability measure over events of the type $\pi_e \leq r(e; \rho, D_n)$ so that we obtain

$$\mathbb{P}[E_n | V_n, \rho] = \mathbb{P}\left[\bigcap_{e \in E_n} [\pi_e \leq r(e; \rho, D_n)] \right] = F_\pi\big(\boldsymbol{r}(D_n; \rho)\big), \tag{6.3}$$

where F_π is the joint CDF over the preference variables π_e. Thus, for a given observation D_n, *any* probability over the set of preference events $\pi_{\alpha\beta} \leq r(e; \rho, D_n)$ will take on the form of a joint CDF $F_\pi\big(\boldsymbol{r}\big)$ over the preference variables $\boldsymbol{\pi} \equiv \{\pi_{\alpha\beta}\}_{(\alpha,\beta) \in E_n}$, where the CDF F_π is evaluated at $\boldsymbol{r}(D_n; \rho)$.

Given multiple independent observations $\mathscr{D} = \{D_1, \ldots, D_N\}$, we can then define a *structured loss functional* $\mathscr{L}(\mathscr{D}; \rho, F_\pi)$ as the log-probability of independently generating the observed orderings in $\mathscr{D}$, so that

$$\mathscr{L}(\mathscr{D}; \rho, F_\pi) \equiv -\sum_{n=1}^{N} \log F_\pi\big(\boldsymbol{r}\big) \tag{6.4}$$

where each term in the loss functional is the log of a joint CDF. Whilst each of these log-CDF terms is defined over many preference variables with a high degree of dependence amongst variables, we can nevertheless represent each term compactly as a cumulative distribution network (CDN) [8, 10], which is a graphical model representing the joint CDF of several random variables (see Appendix). An example of a possible CDN representing a joint CDF over three pairwise preferences is shown in Fig. 6.2c.

Having defined the structured loss functional $\mathscr{L}(\mathscr{D}; \rho, F_\pi)$, the problem of learning to rank sequences from observations $D_1, \ldots, D_N$ will then consist of minimizing the loss functional with respect to the ranking function ρ and the CDF F_π. Let $\boldsymbol{\theta}$ denote the vector of parameters which parameterize both the ranking function ρ and the joint CDF F_π, so that we can write the structured loss as a function of $\boldsymbol{\theta}$, or

$$\mathscr{L}(\mathscr{D}; \boldsymbol{\theta}) \equiv \mathscr{L}(\mathscr{D}; \rho, F_\pi) = \sum_{n=1}^{N} \mathscr{L}(D_n; \boldsymbol{\theta}) = -\sum_{n=1}^{N} \log F_\pi\big(\boldsymbol{r}(D_n; \boldsymbol{\theta})\big). \quad (6.5)$$

In order to optimize $\mathscr{L}(\mathscr{D}; \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, we will assume that we can compute the gradient $\nabla_{\boldsymbol{\theta}} \mathscr{L}(D_n; \boldsymbol{\theta})$ for each observation D_n. Given the gradient, we can then proceed to optimize the structured loss functional using a stochastic gradients descent (SGD) algorithm whereby for each observation D_n, we construct a CDN for order graph G_n and we update the parameters of the model according to the rule $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \mu \nabla_{\boldsymbol{\theta}} \mathscr{L}(D_n; \boldsymbol{\theta})$, where μ is a learning rate parameter for the SGD algorithm. This leads to an efficient method for learning to rank, as we only need to store the CDN for a single observation for the purpose of computing a gradient and updating the model parameters: this is illustrated graphically in Fig. 6.3.

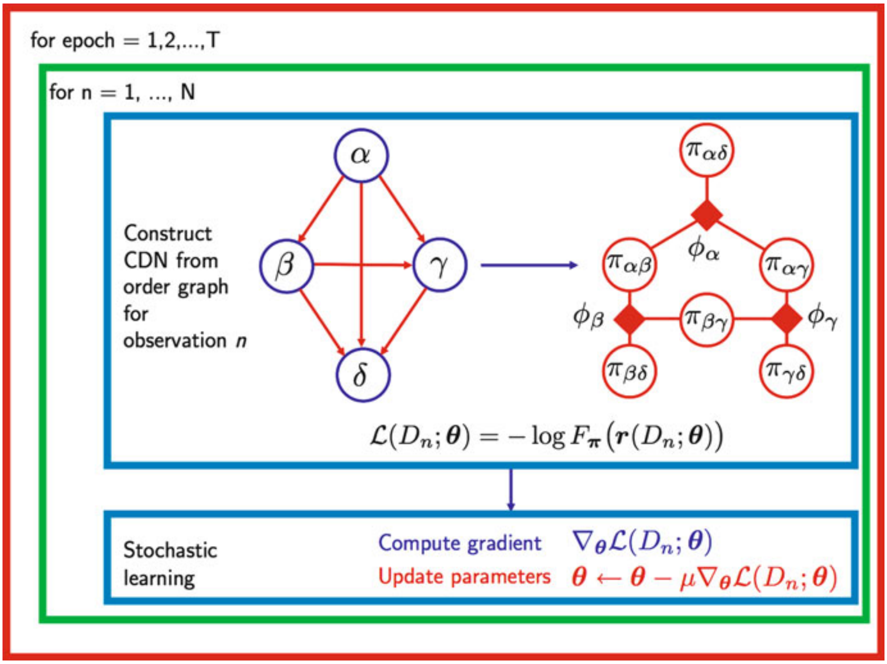
6.2.1 Ranking using Sequence and Quantitative Features

In order to adapt the above framework to the problem of ranking sequences, we will use a ranking function $\rho(\alpha)$ which has the general form

$$\rho(\alpha) = \rho_{seq}(s_\alpha; \mathbf{M}) + \rho_{quant}(\mathbf{x}_\alpha; \mathbf{w}) \quad (6.6)$$

where ρ_{seq}, ρ_{quant} are functions which assign scores to the sequence s_α and its corresponding feature vector $\mathbf{x}_\alpha$. Here, it is possible to specify different parametric forms for $\rho(\alpha)$ which assign scores to sequences under various assumptions. In order to score any given node α based on sequence s_α alone, we will consider the sum of contributions of subsequences of s_α under the assumption that each subsequence contributes independently to the overall score for s_α (see Appendix). We will choose ρ_{quant} to be a linear function of the quantitative features, so that $\rho_{quant}(\mathbf{x}_\alpha; \mathbf{w}) = \mathbf{w}^T \mathbf{x}_\alpha$. Given these parameterizations, a sequence s_α will have a higher score if both ρ_{seq} and ρ_{quant} assign high scores to s_α.

The proposed framework of learning the ranking function from observations is then summarized as follows: given a training set of observations consisting of sequences to be ranked, associated quantitative features and a partial ordering



for epoch = 1,2,...,T
for n = 1, ..., N
Construct CDN from order graph for observation *n*
α
β
γ
δ
$\pi_{\alpha\delta}$
ϕ_α
$\pi_{\alpha\beta}$
$\pi_{\alpha\gamma}$
ϕ_β
$\pi_{\beta\gamma}$
ϕ_γ
$\pi_{\beta\delta}$
$\pi_{\gamma\delta}$
$\mathcal{L}(D_n;\boldsymbol{\theta}) = -\log F_{\boldsymbol{\pi}}(\boldsymbol{r}(D_n;\boldsymbol{\theta}))$
Stochastic learning
Compute gradient $\nabla_{\boldsymbol{\theta}}\mathcal{L}(D_n;\boldsymbol{\theta})$
Update parameters $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \mu\nabla_{\boldsymbol{\theta}}\mathcal{L}(D_n;\boldsymbol{\theta})$

Fig. 6.3 Illustration of the STORMSeq framework. For each observation D_n, we construct a CDN defined over preference variables corresponding to edges in the order graph G_n (*top*). For this example, we have an order graph defined over four nodes and six preference variables. The CDN then models the joint CDF over the six preference variables as a product of functions: here the model consists of a product of three functions so that $F_{\boldsymbol{\pi}}\Big(\mathbf{r}(D_n;\boldsymbol{\theta})\Big) = \phi_\alpha(r_{\alpha,\delta}, r_{\alpha,\beta}, r_{\alpha,\gamma})\phi_\beta(r_{\alpha,\beta}, r_{\beta,\gamma}, r_{\beta,\delta})\phi_\gamma(r_{\alpha,\gamma}, r_{\beta,\gamma}, r_{\gamma,\delta})$. Once the CDN has been constructed, we can perform stochastic learning of parameters by computing the gradient of the log-CDF modeled by the CDN and then updating the vector of parameters $\boldsymbol{\theta}$ (*bottom*). We can then repeat this process for each observation and for a number T of epochs, or passes through the training set

over the sequences, we wish to learn a ranking function $\rho(\alpha)$ which maximizes the probability of generating the observed orderings by assigning higher scores to those sequences which are most consistently highly ranked in the observations $\{D_1, \ldots, D_n\}$. In order to learn $\rho(\alpha)$, we can compute the gradients $\nabla_{\mathbf{M}}\rho_{seq}(s_\alpha; \mathbf{M})$, $\nabla_{\mathbf{w}}\rho_{quant}(\mathbf{x}_\alpha; \mathbf{w})$ (see Appendix) in order to perform gradient-based learning. The ranking function is such that we can account for sequence data in addition to other quantitative features such as expression measurements. The use of CDNs to represent the structured loss functional for learning to rank then allows us to account for the fact that learning to rank is inherently a problem in which one must account for the presence of statistical dependence relationships between model variables.

We emphasize at this juncture that STORMSeq has been formulated in a general way so that it is applicable to many different problems in which we wish to learn a ranking function using multiple instances of orderings, sequence data and other quantitative features. To illustrate how STORMSeq might be used in practice, we will apply it to two problems of sequence search. In the first of these problems, we will score sequences bound by transcription factors using the protein binding microarray data of [3]. In the second, we will score targets of the let-7b microRNA in human retinoblastomas using both microRNA overexpression data [9] and other quantitative features such as protein abundance and mRNA expression levels of targets. Before we proceed, it will be instructive to study the relation between STORMSeq and a previous method for learning to rank sequences from orderings over sequences obtained from microarray measurements.

6.2.2 The RankMotif++ Model as a Cumulative Distribution Network

It is worth noting that in the RankMotif++ model of [5], the objective being minimized corresponds to the log-CDF over preferences under the assumption that preference variables are mutually independent. More precisely, in RankMotif++ the loss function is given by $\mathscr{L}(\boldsymbol{\theta}) = \log F_{\boldsymbol{\pi}}(\mathbf{r}(D_n))$, where the probability over all pairwise preferences $\alpha \succ \beta$ is represented by a product over logistic functions of $r_{\alpha\beta} = \rho(\alpha) - \rho(\beta)$ so that

$$\begin{aligned} F_{\boldsymbol{\pi}}(\mathbf{r}(D_n)) \equiv \mathbb{P}\big[\boldsymbol{\pi} \leq \mathbf{r}(D_n)\big] &= \prod_s \frac{1}{1+\exp(-\nu r_s)} \\ &= \prod_{\alpha \succ \beta} \frac{1}{1+\exp\big(-\nu\big(\rho(\alpha)-\rho(\beta)\big)\big)} \end{aligned} \tag{6.7}$$

with $\rho(\alpha) = \rho_{seq}(s_\alpha)$ and $\nu > 0$. Thus the above loss function can be represented using a disconnected CDN model where each function node corresponds to the CDN function $\phi_s(r_s) = \big(1+\exp(-\nu r_s)\big)^{-1}$ and all pairwise object preferences are modeled as being independent of one another.

6.3 Results

6.3.1 Discovering Transcription Factor Binding Profiles

We will first apply the proposed structured ranking learning framework to the problem of ranking sequences using measurements from a protein binding microarray (PBM) experiment. We obtained PBM data from the Supplementary Material section of [3], which consisted of measured intensities of 35-mer probes bound by five different transcription factors Cbf1, Ceh-22, Oct-1, Rap1, Zif268 across two experimental replicate arrays *Array 1* and *Array 2*. The PBM data consisted of intensity measurements y_α for a set of sequences $\{s_\alpha \in \mathscr{S}\}$, where each probe on the array is indexed by α and s_α denotes the nucleotide sequence of a given probe on the array. We used the array labeled *Array 1* as our training data and the probe measurements from *Array 2* as test data. The goal here is to then learn a ranking function which assigns scores to probe sequences under the assumption that higher scores should indicate an increased probability of a TF binding to a sequence.

We applied the STORMSeq method and evaluated the resulting ranking function on the test set. In order to compare STORMSeq to similar methods, we also ran the MatrixREDUCE [6], MDScan [17], Prego [20] and RankMotif++ methods on the same training data and evaluated these on the same test data using the settings specified by [5] (see Appendix for details). Here we applied STORMSeq without using additional quantitative features to provide a fair comparison to the other methods which rank sequences using only sequence data. The performance of all five methods for the above five TFs are summarized in Fig. 6.4a and 6.4b using precision versus recall curves, as well as Normalized Discounted Cumulative Gain [13] curves which account for how well a method ranks high-intensity sequences (see Appendix). The use of the NDCG metric here is well-suited to the problem at hand, as the truncation level n can be interpreted as the number of sequences to be further validated or analyzed, so that a higher NDCG value is obtained if the most significant sequences appear at the top of the list in their correct order of significance. Here, the significance of a sequence is determined by the strength with which a transcription factor binds to it, so that the highest score should be assigned to the

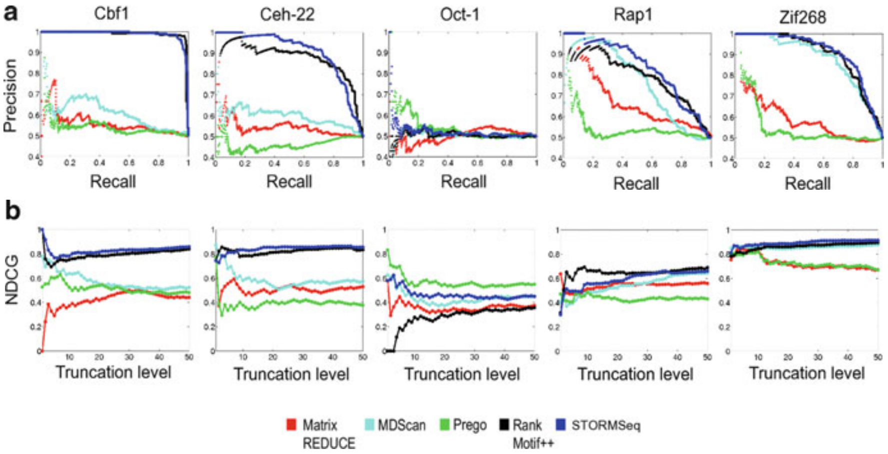


Fig. 6.4 (**a**) Precision versus recall using five different methods for the Cbf1, Ceh-22, Oct-1, Rap1, Zif268 transcription factors studied in [3,5]. The methods shown are MatrixREDUCE (*red*), MDScan (*cyan*), Prego (*green*), RankMotif++ (*black*) and STORMSeq (*blue*); (**b**) The corresponding curves showing Normalized Discounted Cumulative Gains (NDCG) versus the truncation level, or the number of top-ranking sequences. Both (**a**) and (**b**) show that by ranking in a structured learning setting using STORMSeq, we generally improve predictive accuracy, in terms of precision, recall and NDCG, with respect to the other unstructured learning methods shown here

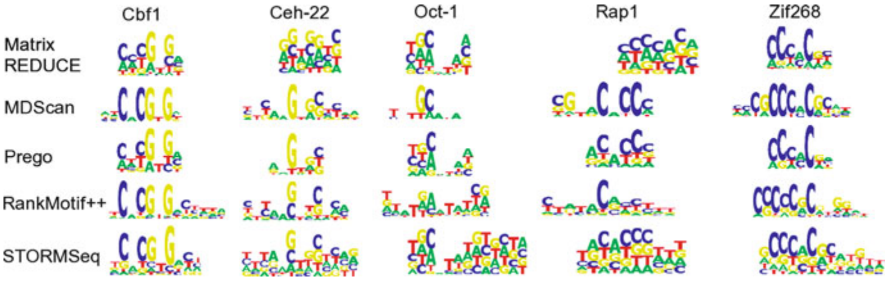


Fig. 6.5 Motifs found by the MatrixREDUCE, MDScan, Prego, RankMotif++ and STORMSeq methods (*rows*) for each of the TFs

most strongly bound sequence. Figure 6.4a and 6.4b demonstrate that by ranking in a structured learning setting and by making no particular assumption about the relationship between sequence s and measured PBM intensities, we increase predictive accuracy as measured by precision, recall and NDCG compared to the other unstructured prediction methods such as RankMotif++. In particular, according to the NDCG metric, our method of ranking also has increased accuracy in terms of the ranking itself, so that sequences with higher intensities are more likely to be ranked higher by STORMSeq than by the other models.

The corresponding PSSMs found by each of the above methods are shown in Fig. 6.5. As can be seen, the PSSMs learned by STORMSeq are consistent with those found by the other methods as well as with PSSMs previously reported for this dataset [3, 5]. It is worth noting here that the consensus sequence for RAP1 found by our method, as well as the consensus reported by the Prego and MDScan methods agree with the First 6 base positions of the widely published motif $ACACCC$ [21]. Also, observe that while the PSSMs obtained by STORMSeq can be degenerate at many positions for various TFs, the improved performance of STORMSeq over these methods suggests that these methods are likely to underestimate the degeneracy of the motifs to be discovered as a consequence of model misspecification.

One reviewer has pointed out that the particular sequence ranking function used above is not designed to allow for gaps in motifs [4]. One advantage of the structured ranking learning framework is that the user can choose from many ranking functions for any given problem, so that the user can specify a ranking function which accounts for the presence of gaps, or other specific features of the motifs to be found. In the case where we wish to learn a PSSM for gapped motifs, we can constrain the degenerate positions in the PSSM by constraining the entropy of the nucleotide frequency at these positions: we provide an example of this in the Appendix.

Having applied the structured ranking learning framework to the problem of discovering transcription factor binding sites, we will also demonstrate the usefulness of STORMSeq for discovering microRNA targets, which also consist of short nucleotide sequences which regulate the activity of genes.

6.3.2 Discovering microRNA Targets

In addition to learning to rank transcription factor binding sites, we will also demonstrate the usefulness of STORMSeq for ranking microRNA targets. MicroRNAs consist of molecules of 22–25 nucleotides which target mRNA transcripts through complementary base-pairing to short target sites, in a fashion analogous to the operation of transcription factors. However, unlike transcription factors, microRNAs are generally inhibitory in their activity, so that microRNA activity generally represses the activity of their target genes either by reducing the abundance of their target mRNA transcripts or by repressing translational activity of their target mRNAs [1, 9]. There is substantial evidence that microRNAs are an important component of the cellular regulatory network, providing a post-transcriptional means to control the amounts of mRNA transcripts and their protein products [1, 7, 9, 14]. As a consequence of their important role in gene regulation, many previous methods have been proposed for performing genome-wide discovery of targets of microRNAs [7, 9, 16, 18].

We will focus here on the let-7b microRNA and a dataset profiling the expression of human mRNAs in WERI-Rb1 retinoblastoma samples after the transfection of a synthetic RNA duplex of the mature let-7b hairpin [9]. Under the assumption that microRNA regulation is causes reduced mRNA expression, pairwise preference relationships between sequences were asserted using the same criteria as in [5], but using negative log-expression-ratios of expression from the let-7b transfections. Thus, the score of a sequence should correspond to the amount of down-regulation by let-7b. We constructed our dataset in a fashion similar to that used in the previous example for transcription factor binding sites (see Appendix). In contrast to the previous problem which had relatively few sources of data variability, here we are provided with *in vivo* expression measurements of genes which may have several different regulators, some of which may themselves be regulated by let-7b. The problem of scoring microRNA targets is therefore representative of the type of problem more commonly encountered in genomics, where the goal is to discover sequences in the presence of many sources of *in vivo* regulatory variability. The hypothesis here is that we can leverage additional information in the form of independent quantitative measurements and computational predictions in order to better account for the variability in orderings over sequences.

To learn to rank microRNA targets, we used human 3′UTR sequence data, mouse mRNA expression, mouse let-7b expression and mouse protein abundance data [1, 15, 22] across brain, heart, liver, lung and placenta tissue pools, whereby the mouse mRNAs were selected as homologs of the human mRNAs in the above WERI-Rb1 assay. Furthermore, the expression for the let-7b microRNA in the above tissue pools corresponds to that of mouse homolog for let-7b (see Appendix). Here we selected sequences which have associated mouse mRNA and protein measurements.

In addition to expression features, we would also like to account for other contextual sequence features, such as microRNA site accessibility. To this end, we ran the PITA [14] algorithm for computing an accessibility score for each 3′UTR sequence given the mature let-7b sequence. This score, which we will here denote as $\Delta\Delta G$, is a function of the accessibility of a target site given the most likely secondary structure of the target mRNA. Combined with the above mRNA, microRNA and protein abundance features, this yielded a total of 16 quantitative features for each sequence to be scored. Thus for this problem, each $3'$UTR sequence corresponds to a putative let-7b-target interaction so that let-7b putatively targets at least one target site in the $3'$UTR sequence. The above 16 features thus form the feature vector $\mathbf{x}_\alpha$ which we will use for learning to rank microRNA targets.

6.3.2.1 Incorporating Diverse Computational Predictions

In addition to the above features, we would like to also incorporate computational target predictions for let-7b from the PicTar [16], TargetScan [7] and RNA22 [18] sequence-based target prediction methods. In order to assign scores to candidate microRNA targets, each of these methods makes use of various criteria such as conservation and contextual sequence features. The scores output by these prediction methods can be then used to generate an order graph over sequences, so that each method provides a partial ordering over some subset of microRNA-target interactions (see Appendix).

Given all of the above, we applied STORMSeq under three settings, where (a) we only used sequence data for learning to rank targets, (b) we only used quantitative features (mRNA and microRNA expression, protein abundance and $\Delta\Delta G$), and (c) we also used information provided by diverse computational prediction methods in addition to both sequence and quantitative features (see Appendix). To assess the out-of-sample predictive performance of our method, we selected a random sample of 250 positive sequences for our training data and the remainder for the test data. Similarly, we selected 250 sequences from the negative group for our training set and the rest for the test data. We thus formed five independent training/test splits in this fashion (see Appendix). For each of the five train/test datasets, we computed precision and recall for each of these experimental settings. The resulting precision and recall curves, averaged over the five test sets, are shown in Fig. 6.6. As can be seen, incorporating sequence data, quantitative features and computational predictions together under one model yields an improvement in predictive accuracy compared to using sequence alone or sequence in tandem with quantitative features. This indicates that by leveraging multiple sources of information about microRNA regulation, we can significantly increase the accuracy with which we discover microRNA targets.

For further validation, we show the cumulative distribution of $\Delta\Delta G$ scores for the top and bottom 100 targets ranked according to STORMSeq (Fig. 6.7a). We expect *a priori* that sequences with lower $\Delta\Delta G$ score are more likely to be bound by a targeting microRNA than not. As can be seen, high-scoring targets have a significantly lower average $\Delta\Delta G$ value than low-scoring targets ($P < 10^{-20}$, Wilcoxon-Mann-Whitney test), demonstrating that the targets discovered by STORMSeq are likely to be genuinely targeted by let-7b. Furthermore, the

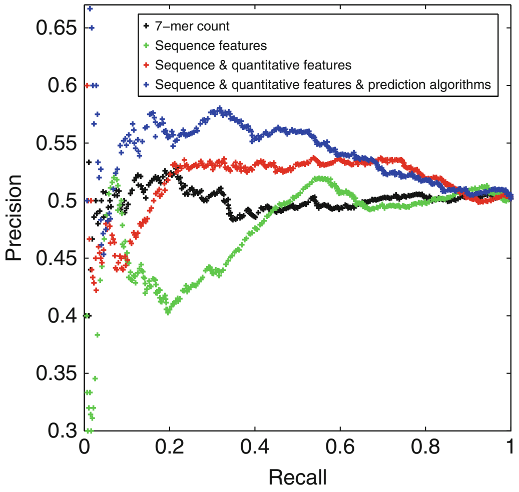


Fig. 6.6 Precision versus recall for different STORMSeq learning configurations using expression data for mRNAs in response to let-7b transfection [9]. By incorporating additional sources of sequence information, sequence context and quantitative profiling features, STORMSeq achieves higher accuracy (*blue*) than using 7-mer counts to predict downregulation (*black*), using sequence data alone (*green*) or sequence data combined with quantitative features without computational predictions as additional data (*red*)

protein abundances for the top and bottom 100 targets differed significantly as well (Fig. 6.7b, $P = 7.73 \times 10^{-4}$), adding support for the hypothesis that the targets which receive a high score under STORMSeq are *bona fide*, as microRNA activity generally leads to lower protein abundance and mRNA transcript abundance [1, 7, 9, 14].

To assess the use of purely sequence-based methods for this problem, we also ran the MEME [2] and AlignACE [12] algorithms using default settings on the 250 positive sequences for each training set and examined the resulting PSSMs reported by both algorithms. The PSSMs obtained from these methods can then be used to rank sequences. We found that for all five training/test datasets, none of the PSSMs discovered by MEME and AlignACE led to any significant ability to rank let-7b targets (data not shown), suggesting that without additional information in the form of sequence conservation or quantitative measurements, *de novo* approaches to scoring sequences are significantly more likely to find poor models by virtue of either using only sequence information or by virtue of model misspecification.

6.4 Discussion

We have presented the STORMSeq method for learning to rank regulatory sequences by combining heterogeneous datasets and diverse computational prediction methods. The explicit formulation of sequence search as a problem of ranking

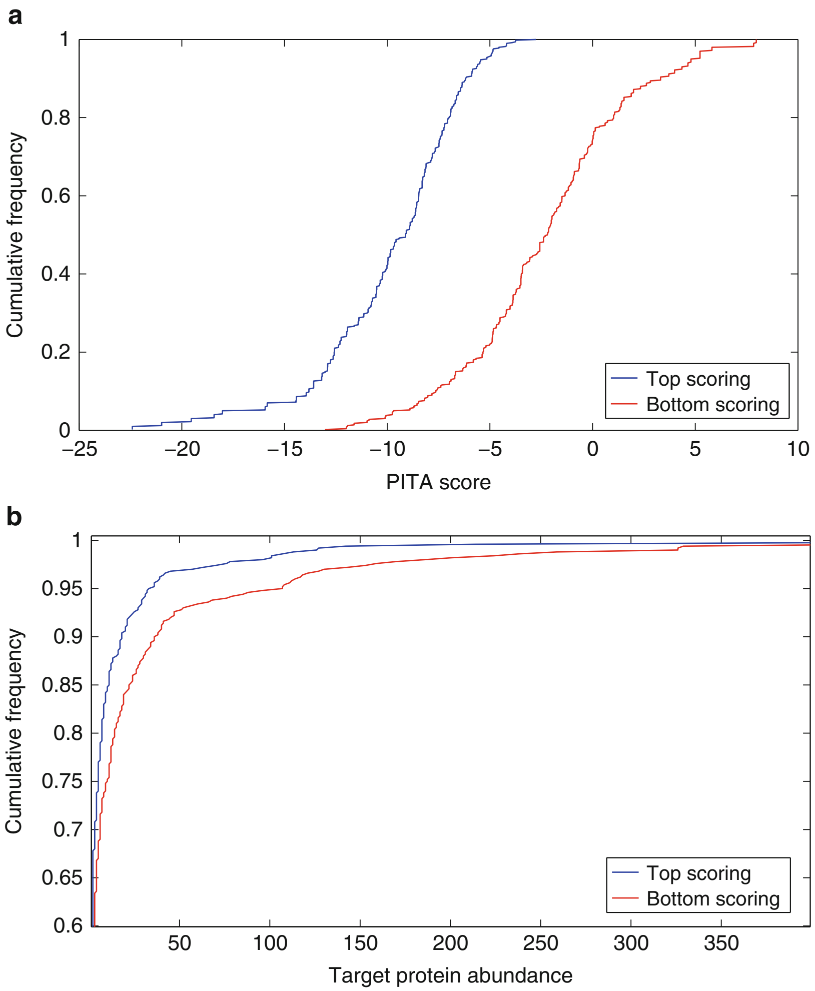


Fig. 6.7 (**a**) Cumulative frequency plots of the $\Delta\Delta G$ scores on the top and bottom 100 targets as ranked by STORMSeq. High-scoring STORMSeq targets generally have higher target site accessibility and so have a lower $\Delta\Delta G$ value compared to low-scoring targets ($P < 10^{-20}$, Wilcoxon-Mann-Whitney); (**b**) Cumulative frequency plots of protein abundances for top and bottom 100 targets as ranked by STORMSeq. High-scoring STORMSeq targets have significantly lower target protein abundance ($P = 7.73 \times 10^{-4}$) as a result of microRNA repressive activity

accounts for the fact that different sequences can have multiple levels of significance and any method for ranking should correctly order sequences by assigning a high score to biologically significant sequences. In particular, by accounting for the statistical dependence relationships which exist in learning to rank, STORMSeq improves predictive performance over other unstructured methods for learning to rank. In addition, STORMSeq largely avoids many of the issues of model misspecification and complex inference which may arise when modelling multiple heterogeneous datasets. As STORMSeq is formulated in fairly general terms, it can also be applied to other problems of sequence search such as ranking drug targets, discovering genetic associations or scoring protein-protein interactions, although we have not focused on such applications here.

In the case of ranking microRNA-target interactions we have shown that incorporating diverse computational predictions increases predictive accuracy as measured by precision and recall. It should be noted that one must exercise care in what additional sources of computational predictions are incorporated into the analysis. We found that by incorporating computational prediction methods which had inherently low accuracy, we could in fact decrease the predictive accuracy of our method (data not shown). In our case, particular computational prediction methods were included in our analysis on the basis of a previous study conducted in [9] which gauged the predictive accuracy of a variety of microRNA-target prediction methods according to a variety of metrics. We caution that in the case in which data is relatively limited in size, including computational predictions from methods which have low accuracy can adversely impact the accuracy of STORMSeq. A possible extension to the framework proposed here is to allow for outlier detection so that the model can discount the impact of outlier observations.

One reviewer pointed out that the optimization problem being solved is generally non-convex and may assign high probability to different orderings over sequences. Although the underlying ranking may not be unique for a given class of ranking functions and/or loss functionals, there may be a large number of *partial* orderings over sequences which are consistent with an underlying (and possibly unidentifiable) total ordering over sequences. Thus, although many orderings may be possible and STORMSeq may learn one of these, those which are most useful in practice are those orderings in which the *relevant* sequences are correctly ranked, while less of a penalty should be assigned whether we have correctly ranked the less relevant sequences. Thus the issue of whether the ranking of relevant sequences is identifiable may be of concern, so that standard techniques for avoiding poor local minima must be used and the solutions obtained from multiple restarts should be compared with one another.

An important issue which arises often in practice concerns the tractability of the proposed framework. In a setting in which one is given a large number of sequences to be ranked for a single observation, the number of edges in an order graph may in the worst case reach $O(n^4)$, where n is the number of objects in the observation. As storing and processing such a large observation may be intractable, we have made use of the mean absolute deviation (MAD) criterion for asserting preference relationships (see Appendix), which has the effect of reducing the number of pairwise preferences to be modeled. One can devise similar schemes to reduce the number of pairwise preferences to be modeled, as many of these will represent pairwise ordering constraints between very highly relevant sequences and irrelevant ones. We have also found that one can randomly break up an observation defined over many sequences into a set of multiple observations defined over smaller subsets of the sequences. Each of these observations could then be tractably modeled using the proposed method. In addition or as an alternative to the above, one can choose a CDN graph which is tractable and amenable to fast computations. An advantage of the proposed framework is that it is possible to use sparser CDN graphs which tradeoff the presence of dependencies between pairwise preferences for tractability and speedups in computation time.

We have applied STORMSeq to the problems of scoring sequences bound by transcription factors and scoring microRNA targets, whereby performing structured learning and combining different data types with computational predictions was shown to improve predictive accuracy. In the case of ranking microRNA targets, features relating to expression patterns in mouse proved to increase the ranking accuracy of scoring targets in human retinoblastomas. This suggests that STORMSeq may also be useful for problems in comparative genomics as a principled means for combining diverse datasets from different species. Other interesting extensions of the STORMSeq would include scaling the proposed framework to genome-wide detection of regulatory sequences as well as using richer representations for the ranking function which could account for direct interactions between the sequences to be ranked.

Acknowledgements BJF is a Fellow of the Canadian Institute for Advanced Research and holds a Canada Research Chair in Information Processing and Machine Learning. This research was partly funded by CIHR and OGI/Genome Canada.

Appendix

Cumulative Distribution Networks

The CDN [8, 10] is an undirected bipartite graphical model in which the joint CDF $F(\mathbf{z})$ over a set of random variables is modeled as a product over functions defined over subsets of these variables. More formally, for variable set $\mathbf{Z}$, the joint CDF is given by

$$F(\mathbf{z}) = \prod_{s \in S} \phi_s(\mathbf{z}_s), \tag{6.8}$$

where S is a set of function indices and for $s \in S$, $\phi_s(\mathbf{z}_s)$ is defined over some subset of the variables in $\mathbf{Z}$. For detailed derivations of the properties of CDNs, including marginal and conditional independence properties, we refer the reader to [10]. The CDN framework provides us with a means to compactly represent multivariate joint CDFs over many variables: in the next section we will formulate a loss functional for learning to rank which takes on such a form.

A Structured Loss Functional for Learning to Rank

Let the ranking function $\rho(\alpha) \equiv \rho(\alpha; \mathbf{a})$ be parameterized by the parameter vector $\mathbf{a}$ so that $\boldsymbol{r}(D_n; \rho) \equiv \boldsymbol{r}(D_n; \mathbf{a})$. For a given order graph G_n, the structured loss functional is then given by

$$\mathscr{L}(D_n;\boldsymbol{\theta}) \equiv \mathscr{L}(D_n;\mathbf{a},\nu) = -\log F_\pi\big(\boldsymbol{r}(G_n;\mathbf{a})\big) = -\log\phi(\mathbf{r}(D_n;\mathbf{a})) \tag{6.9}$$

where $\boldsymbol{\theta} = \big[\mathbf{a}\ \nu\big]$ is the set of parameters. Here we can choose from a wide variety of CDN topologies and functional forms for the CDN functions, such as the particular CDN used in [11]. We will represent the joint CDF using a single CDN function $\phi(\mathbf{r})$ set to a multivariate sigmoidal function so that

$$\phi(\mathbf{r}) = \frac{1}{1+\sum_e \exp(-\nu r(e;\mathbf{a},D_n))}, \quad \nu > 0. \tag{6.10}$$

For the given CDN and ranking functions, the learning problem for the current observation D_n then becomes

$$\min_{\mathbf{a},\nu} \quad \sum_n \log\left(1 + \sum_{e\in E_n} \exp\big(-\nu r(e;\mathbf{a},D_n)\big)\right) \quad \text{s.t.} \quad \nu > 0. \tag{6.11}$$

In order to solve the above optimization problem, we will use a stochastic gradient descent algorithm which will require us to compute the gradient $\nabla_{\boldsymbol{a}}\mathscr{L}(D_n;\boldsymbol{\theta})$ for each observation D_n. This is given by

$$\nabla_{\mathbf{a}}\mathscr{L}(D_n;\boldsymbol{\theta}) = \nu\phi\big(\mathbf{r}(D_n;\mathbf{a})\big)\sum_{e\in E_n}\exp(-\nu r(e;\mathbf{a},D_n))\nabla_{\mathbf{a}} r(e;\mathbf{a},D_n),$$

with

$$\nabla_{\mathbf{a}} r(e;\mathbf{a},D_n) = \nabla_{\mathbf{a}}\rho(\alpha;\mathbf{a}) - \nabla_{\mathbf{a}}\rho(\beta;\mathbf{a}) \tag{6.12}$$

The derivative with respect to the CDN function weight w is then given by

$$\partial_\nu\Big[\mathscr{L}(D_n;\boldsymbol{\theta})\Big] = -\sum_{e\in E_n} r(e;\mathbf{a},D_n)\exp\big(-\nu r(e;\mathbf{a},D_n)\big)\ \phi(\mathbf{r}) \tag{6.13}$$

With the above gradients, we can then proceed to construct a CDN for each observation D_n and updating the parameters of the model according to the rule $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \mu\nabla_{\boldsymbol{\theta}}\mathscr{L}(D_n;\boldsymbol{\theta})$, where μ is a learning rate parameter.

Ranking Functions for Sequence and Quantitative Features

Suppose we are given a sequence s_α of length L_α which we would like to score. Let $s_\alpha^{k:k+K-1}$ be a subsequence of s_α of length K starting at position k and let s_α^j be the symbol observed at position j in sequence s_α. Given a PSSM $\mathbf{M}$ of length K (where

$M_{k,b}$ is equal to the probability of emitting symbol b at position k of the PSSM) we can define the score for sequence s_α as the probability that a transcription factor binds to at least one subsequence of length K in s_α according to the PSSM, so that

$$\rho_{seq}(s_\alpha;\mathbf{M}) = \log\Big(1 - \prod_{k=0}^{L_\alpha - K} (1 - P(s_\alpha^{k+1:k+K}|\mathbf{M}))\Big) \tag{6.14}$$

where $P(s_\alpha^{k+1:k+K}|\mathbf{M}) = \prod_{j=k+1}^{k+K} M_{j,s_\alpha^j}$ is the probability of binding to subsequence $s_\alpha^{k+1:k+K}$ according to $\mathbf{M}$. The derivative of the ranking function $\rho_{seq}(s_\alpha;\mathbf{M})$ with respect to the parameter $M_{k,b}$ is equal to

$$\frac{\partial \rho_{seq}(s_\alpha;\mathbf{M})}{\partial M_{k,b}} = \frac{1 - \exp\big(\rho_{seq}(s_\alpha;\mathbf{M})\big)}{\exp\big(\rho_{seq}(s_\alpha;\mathbf{M})\big)} \times \left(\sum_i \frac{P(s_\alpha^{i+1:i+K}|\mathbf{M})}{1 - P(s_\alpha^{i+1:i+K}|\mathbf{M})}\big([s_\alpha^{i+k} = b] - P(b|\mathbf{M})\big)\right) \tag{6.15}$$

We can then collect these derivatives into a vector to form the gradient $\nabla_{\mathbf{M}}\rho_{seq}(s_\alpha;\mathbf{M})$.

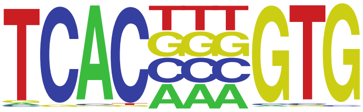
In the case where we are provided with quantitative features in the form of a feature vector $\mathbf{x}_\alpha$, we can define the ranking function $\rho_{quant}(\mathbf{x}_\alpha;\mathbf{w})$ to be a linear function given by $\nabla_{\mathbf{w}}\rho_{quant}(\mathbf{x}_\alpha;\mathbf{w}) = \mathbf{x}_\alpha$. Once we have computed both gradients, we can evaluate

$$\nabla_{\mathbf{a}}\rho(\alpha;\mathbf{a}) = \begin{bmatrix} \nabla_{\mathbf{M}}\rho_{seq}(s_\alpha;\mathbf{M}) \\ \nabla_{\mathbf{w}}\rho_{quant}(\mathbf{x}_\alpha;\mathbf{w}) \end{bmatrix}. \tag{6.16}$$

Ranking Functions for Discovering Gapped Motifs

In the case in which we wish to allow for gaps, we can posit a ranking function of the same form as in Eq. 6.14, but with an additional constraint that for degenerate positions j in the PSSM $\mathbf{M}$, we have $M_{j,a} = 0.25$ for $a \in \{A, C, G, T\}$. This constraint is equivalent to forcing certain positions to be contribute the same score to the total sequence score regardless of what nucleotides occur at these positions. Alternatively, we could regularize each degenerate position of the PSSM by adding some constant C_j to each entry $M_{j,a}$, where C_j is chosen so that for position j is a distribution that is close to being uniform. For the former constraint, we would simply update the entries of the PSSM for only non-degenerate positions. For the latter constraint, we can regularize the appropriate entries of $\mathbf{M}$ during the learning process by simply adding C_j after each update of the PSSM. An example of the PSSM for such a gapped motif is shown in Fig. 6.8. It is worth noting that the length of the gap, or number of degenerate positions in the PWM, can either be specified by the user or it can be selected via cross-validation, as with the length of the PWM.

Fig. 6.8 An example of a gapped motif



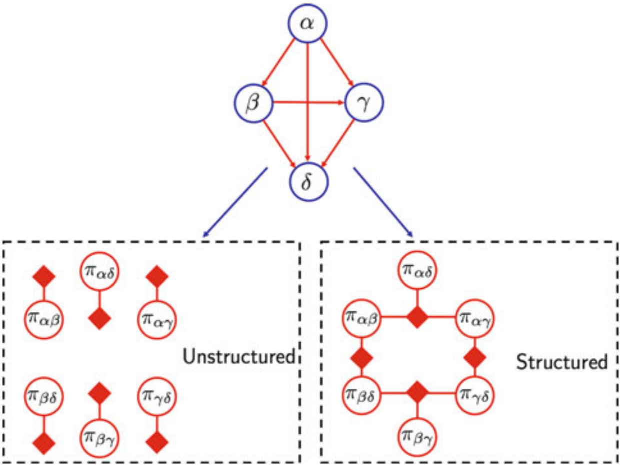


Fig. 6.9 An example of an order graph over four nodes $\alpha, \beta, \gamma, \delta$ corresponding to the ordering $\alpha \succ \beta \succ \gamma \succ \delta$, with CDNs representing two different loss functions corresponding to different independence assumptions about pairwise preferences. Whereas the RankMotif++ method of [5] corresponds to an unstructured learning method which assumes independence of preference variables, STORMSeq models the dependencies between preferences by introducing connections between preference variables in the corresponding CDN

The RankMotif++ Method as a Disconnected CDN

For the RankMotif++ model of [5], the corresponding probability over all pairwise preferences $\alpha \succ \beta$ is modeled by a product over logistic functions of $\rho(\alpha) - \rho(\beta)$ so that $F_\pi(\mathbf{r}(D_n)) \equiv \mathbb{P}[\boldsymbol{\pi} \le \mathbf{r}(D_n)] = \prod_s \frac{1}{1+\exp(-\nu r_s)} = \prod_{\alpha \succ \beta} \frac{1}{1+\exp\big(-\nu\big(\rho(\alpha)-\rho(\beta)\big)\big)}$ with $\rho(\alpha)$ corresponding to the sequence ranking function ρ_{seq} above. This can thus be represented as a completely disconnected CDN where each function node corresponds to $\phi_s(r_s) = \frac{1}{1+\exp(-\nu r_s)}$ and all pairwise object preferences are modeled as being independent of one another. This is illustrated in Fig. 6.9 for an example with four sequences $s_\alpha, s_\beta, s_\gamma, s_\delta$ to be ranked in which we represent the corresponding joint CDF using two different CDNs.

Settings for STORMSeq

We ran STORMSeq for 100 epochs, or passes through the training observations, using a stochastic gradients optimization method. The learning rate was set to $\mu = 0.1$ with a decay rate of $1/t$ at the end of each epoch t. In order to provide regularization on the CDN width parameter ν, we set a constraint $\nu \leq 1$. In the case where we learn a PSSM $\mathbf{M}$, we enforce the constraints that $M_{k,b} > 0 \ \forall \ k, b$ and $\sum_b M_{k,b} = 1 \ \forall k = 1, \ldots, K$. In the case where we learn weights $\mathbf{w}$, we set an additional L_1-norm constraint of $\|\mathbf{w}\|_1 \leq 50$. All computational runs were performed in triplicate and the best optimum achieved on training data was selected for evaluation on test data using criteria described in the sections below. Additional details on the learning method are provided in [11].

Methods for Ranking Sequences Bound by Transcription Factors

Data was downloaded from the Supplementary Material section of [3], which consisted of measured intensities y_α for a set of sequences $\{s_\alpha \in \mathscr{S}\}$. The dataset contained five experiments across two microarrays (*Array 1* and *Array 2*) profiling the binding of the transcription factors Cbf1, Ceh-22, Oct-1, Rap1, Zif268. We used the array labeled *Array 1* as the source of our training data, and the probe sequences from *Array 2* as the source of our test data. We normalized the microarray intensity data in both sets by first shifting microarray intensities such that the minimum intensity was equal to one, then applying a log-transformation, as in [5]. We labelled the 250 probe sequences which had the highest measured intensity as positives and the 250 sequences with the lowest normalized intensities as negatives. We then constructed the order graph over these 500 sequences based on preferences assessed using the criteria used by [5] where we compute the median absolute deviation m of the 500 normalized intensities and asserted $\alpha \succ \beta$ if $y_\alpha > y_\beta + 3\sigma$ and at least one of s_α, s_β were labelled as positive sequences as described above, where $\sigma = m/0.6745$, where 0.6745 is the median absolute deviation of the standard normal.

Using the above sequence ranking function $\rho(s_\alpha; \mathbf{M})$ for a given PSSM length K, we ran STORMSeq and RankMotif++ using three random initializations each, whereby we selected the model which maximized the Spearman correlation with the training data, as per [5]. For each initialization, the PSSM $\mathbf{M}$ was initialized to a set of random positive values and then normalized so that $\sum_b M_{k,b} = 1 \ \forall \ k = 1, \ldots, K$. The MatrixREDUCE, MDScan and Prego methods were run on the training data as specified in [5], and the resulting PSSM models were selected using the same Spearman correlation metric as above. For all models, we varied K from 7 to 13 and selected the value of K which optimized the above Spearman correlation criteria.

Methods for Ranking microRNA Targets

We focused on the human genes in the let-7b transfection experiment which (a) had 3′UTR sequence data provided by Ensembl and (b) were provided with both mRNA expression and protein abundance data in 3,636 paired mRNA-protein expression profiles obtained from cDNA microarray and mass-spectrometry across brain, heart, liver, lung and placenta tissue pools in mouse [15, 22]. This yielded a total of 799 human 3′UTR sequences to be scored. We then selected the 400 sequences with the lowest log-expression ratios as positives and labelled the other 399 genes as negatives. To assess the out-of-sample predictive performance of our method, we selected a random sample of 250 positive sequences for our training data and the remainder for the test data. Similarly, we selected 250 sequences from the negative group for our training set and the rest for the test data. We thus formed five independent training/test splits in this fashion. Preferences were then assessed as described above for the PBM data. Once we obtained the training and test datasets, we ran STORMSeq with $K = 7$ on each of the training datasets and selected the best model out of three random restarts via the Spearman correlation between the learned ranking function scores and the rankings seen in the training data.

In conjunction with the above data, we used the expression let-7b across brain, heart, liver, lung and placenta tissue pools [1] with the mRNA/protein profiles mentioned above. Additionally, the $\Delta\Delta G$ accessibility score was computed by the PITA algorithm [14] using the mRNA sequences for each of the mouse mRNAs in the data from [22] and using the mature mouse let-7b sequence for the default algorithm settings provided in [14].

We downloaded microRNA target predictions for the let-7b microRNA from the Supplementary Data resources for the TargetScan [7], PicTar [16] and RNA22 [18] algorithms. The set of TargetScan predictions contains both conserved and non-conserved targets and the set of RNA22 targets contains both target predicted from 5′UTR and 3′UTR sequences. We mapped all predictions to the above mouse mRNA and microRNA labels. Pairwise preference relationships were established for a given 3′UTR sequence by summing over microRNA target site scores within the given 3′UTR sequence and sorting scores. For a given prediction method, the preference $\alpha \succ \beta$ was established between two 3′UTR's s_α, s_β if s_α had a higher score than s_β and at least one of s_α, s_β were labelled as positive sequences as described above.

Assessing Ranking Performance

To assess predictive performance of any given ranking method, we scored each node α using the ranking function $\rho(\alpha)$ learned by the method. Given the ordering obtained from ρ and given positive/negative labels for the nodes being ranked, we can then compute Precision and Recall as

$$Precision = \frac{TP}{TP + FP}$$
$$Recall = \frac{TP}{TP + FN}$$

where *TP, FP, FN* correspond to the number of true positives, false positives and false negatives respectively.

We also used the Normalized Discounted Cumulative Gains [13] metric, which is commonly in use in information retrieval research. The NDCG accounts for the fact that highly relevant sequences should be ranked higher by a given method, so more weight should be placed on correctly ranking highly relevant sequences than marginally relevant ones. The formula for computing the NDCG for truncation level n, or the number of top-ranking sequences, is

$$NDCG(n) = Z_n \sum_{j=1}^{n} \frac{2^{r(j)} - 1}{c_j} \tag{6.17}$$

where $r(j)$ is an observed label indicating the level of importance of the sequence (e.g.: amount of downregulation from a microRNA) and Z_n is a constant to ensure that $NDCG(n) = 1$ for the perfect ranking, so that higher NDCG indicates increased ability to predict the ordering of sequences. The weights $\{c_1, \ldots, c_n\}$ are an increasing sequence of real-valued positive numbers which allow us to penalize errors made in the top of the ranked list whilst discounting errors made for less relevant sequences. Here we chose $c_j = \log_2(1 + j) \; \forall j = 1, \ldots, n$. The advantage of the NDCG metric is that it does not assume that sequences are to be classified as positive or negative and it accounts for both multiple label values and the fact that highly important sequences should be ranked first. This contrasts with the use of Area Under the ROC Curve, or AUC, which weighs misranking errors equally regardless of where they occur in a ranked list. The NDCG can be also seen as an approximation to the cost of experimentally validating or analyzing sequences at the top of the list which are not biologically relevant.

In the case of where we are scoring sequences bound by transcription factors, we set the labels $r(j)$ to be the normalized array intensities, shifted to be non-negative and scaled to obtain a maximum label of 1. For the purpose of evaluating on let-7b targets, we set the above relevance labels to be the negative log-expression-ratios of each putative target, shifted to be non-negative and scaled to obtain a maximum label of 1.

References

1. Babak, T., Zhang, W., Morris, Q. D., Blencowe, B. J., & Hughes, T. R. (2004). Probing microRNAs with microarrays: Tissue specificity and functional inference. *RNA*, *10*, 1813–1819.

2. Bailey, T. L., Williams, N., Misleh, C., & Li, W. W. (2006). MEME: Discovering and analyzing DNA and protein sequence motifs. *Nucleic Acids Research*, *34*, W369–W373.
3. Berger, M. F., Philippakis, A. A., Qureshi, A. M., He, F. S., Estep, P. W., III, & Bulyk, M. L. (2006). Compact, universal DNA microarrays to comprehensively determine transcription factor binding specificities. *Nature Biotechnology*, *24*, 1429–1435.
4. Chen, C. Y., Tsai, H. K., Hsu, C. M., Chen, M. J., Hung, H. G., Huang, G. T. W., & Li, W. H. (2008). Discovering gapped binding sites of yeast transcription factors. *Proceedings of the National Academy of Sciences*, *105*, 2527–2532.
5. Chen, X., Hughes, T. R., & Morris, Q. D. (2007). RankMotif++: A motif-search algorithm that accounts for relative ranks of K-mers in binding transcription factors. *Bioinformatics*, *23*, i72–i79.
6. Foat, B. C., Morozov, A. V., & Bussemaker, H. J. (2006). Statistical mechanical modeling of genome-wide transcription factor occupancy data by MatrixREDUCE. *Bioinformatics*, *22*, e141–e149.
7. Grimson, A., Farh, K. H., Johnston, W., Garrett-Engele, P., Lim, L., & Bartel, D. (2007). MicroRNA targeting specificity in mammals: Determinants beyond seed pairing. *Molecular Cell*, *27*, 91–105.
8. Huang, J. C. (2009). Cumulative distribution networks: Inference, estimation and applications of graphical models for cumulative distribution functions. Ph.D. thesis, University of Toronto, Toronto, ON, Canada.
9. Huang, J. C., Babak, T., Corson, T. W., Chua, G., Khan, S., Gallie, B. L., Hughes, T. R., Blencowe, B. J., Frey, B. J., & Morris, Q. D. (2007). Using expression profiling to identify human microRNA targets. *Nature Methods*, *4*, 1045–1049.
10. Huang, J. C., & Frey, B. J. (2008). Cumulative distribution networks and the derivative-sumproduct algorithm. In *Proceedings of the twenty-fourth conference on Uncertainty in Artificial Intelligence (UAI)*, Helsinki, Finland.
11. Huang, J. C., & Frey, B. J. (2009). Structured ranking learning using cumulative distribution networks. *Advances in Neural Information Processing Systems (NIPS), 21*, 697–704.
12. Hughes, J. D., Estep, P. W., III, Tavazoie, S., & Church, G. M. (2000). Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae. *Journal of Molecular Biology*, *296*, 1205–1214.
13. Jarvelin, K., & Kekalainen, K. (2002). Cumulated evaluation of IR techniques. *ACM Information Systems*, *20*, 422–446.
14. Kertesz, M., Iovino, N., Unnerstall, U., Gaul, U., & Segal, E. (2007). The role of site accessibility in microRNA target recognition. *Nature Genetics*, *39*, 1278–1284.
15. Kislinger, T., Cox, B., Kannan, A., Chung, C., Hu, P., Ignatchenko, A., Scott, M. S., Gramolini, A. O., Morris, Q., Hallett, M. T., Rossant, J., Hughes, T. R., Frey, B., & Emili, A. (2006). Global survey of organ and organelle protein expression in mouse: Combined proteomic and transcriptomic profiling. *Cell*, *125*, 173–186.
16. Krek, A., Grun, D., Poy, M. N., Wolf, R., Rosenberg, L., Epstein, E. J., MacMenamin, P., da Piedade, I., Gunsalus, K. C., Stoffel, M., & Rajewsky, N. (2005). Combinatorial microRNA target predictions. *Nature Genetics*, *37*, 495–500.
17. Liu, X. S., Brutlag, D. L., & Liu, J. S. (2002). An algorithm for finding proteinDNA binding sites with applications to chromatin-immunoprecipitation microarray experiments. *Nature Biotechnology 20*, 835–839.
18. Miranda, K., Huynh, T., Tay, Y., Ang, Y. S., Tam, W. L., Thomson, A., Lim, B., & Rigoutsos, I. (2006). A pattern-based method for the identification of microRNA binding sites and their corresponding heteroduplexes. *Cell*, *126*, 1203–1217.
19. Segal, E., Raveh-Sadka, T., Schroeder, M., Unnerstall, U., & Gaul, U. (2008). Predicting expression patterns from regulatory sequence in drosophila segmentation. *Nature*, *451*, 535–540.
20. Tanay, A. (2006). Extensive low-affinity transcriptional interactions in the yeast genome. *Genome Research*, *16*, 962–972.

21. Wingender, E., Knüppel, R., Dietze, P., & Karas, H. (1995). TRANSFAC® database as tool for the recognition of regulatory genomic sequences. In H. A. Lim & C. R. Cantor (Eds.), *Bioinformatics & Genome Research* (pp. 275–282). New Jersey: World Scientific Publishing Co., Inc.
22. Zhang, W., Morris, Q. D., Chang, R., Shai, O., Bakowski, M. A., Mitsakakis, N., Mohammad, N., Robinson, M. D., Zirngibl, R., Somogyi, E., Laurin, N., Eftekharpour, E., Sat, E., Grigull, J., Pan, Q., Peng, W. T., Krogan, N., Greenblatt, J., Fehlings, M., van der Kooy, D., Aubin, J., Bruneau, B. G., Rossant, J., Blencowe, B. J., Frey, B. J., & Hughes, T. R. (2004). The functional landscape of mouse gene expression. *Journal of Biology*, *3*(5), 21–43.

Chapter 7
Mixture Tree Construction and Its Applications

Grace S.C. Chen, Mingze Li, Michael Rosenberg, and Bruce Lindsay

Abstract A new method for building a gene tree from Single Nucleotide Polymorphism (SNP) data was developed by Chen and Lindsay (Biometrika 93(4):843–860, 2006). Called the mixture tree, it was based on an ancestral mixture model. The sieve parameter in the model plays the role of time in the evolutionary tree of the sequences. By varying the sieve parameter, one can create a hierarchical tree that estimates the population structure at each fixed backward point in time. In this chapter, we will review the model and then present an application to the clustering of the mitochondrial sequences to show that the approach performs well. A simulator that simulates real SNPs sequences with unknown ancestral history will be introduced. Using the simulator we will compare the mixture trees with true trees to evaluate how well the mixture tree method performs. Comparison with some existing methods including neighbor-joining method and maximum parsimony method will also be presented in this chapter.

7.1 Introduction

There are two major families of methods for building phylogenetic trees: character-based and distance-based. For the character-based methods, the Maximum Parsimony (MP), the method of Maximum Likelihood (ML), and Bayesian methods are the most well-known ones.

G.S.C. Chen (✉) and M. Li
School of Math & Stat, Arizona State University, Tempe, U.S.A.
e-mail: scchen@math.asu.edu, mingzeli@asu.edu

M. Rosenberg
School of Life Sciences, Arizona State University, Tempe, U.S.A.
e-mail: msr@asu.edu

B. Lindsay
Department of Statistics, Penn State University, University Park, U.S.A.
e-mail: bgl@psu.edu

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_7,

© Springer-Verlag Berlin Heidelberg 2011

Among these methods, the Parsimony method was introduced by Edwards and Cavalli-Sforza [3], and is one of the first methods to be used to infer phylogeny. A phylogeny having fewer changes to account for the way a group of sequences has evolved is preferable. In other words, the most parsimonious explanation for the observed data is sought. In the method of Maximum Parsimony [2] the tree with the shortest branch lengths is the best. The steps to create this tree are as follows. First, informative sites, or sites where at least two different states occur in at least two taxa, are identified. A subset of trees (or all trees for less than a dozen taxa) is evaluated using a heuristic approach, and the tree with the shortest branch length is chosen.

For cases where there are large amounts of evolutionary changes in different branches of a tree, the method of Maximum Likelihood (ML) is to be preferred. Maximum Likelihood was created by Ronald A. Fisher [6–8] and later applied to gene frequency data for phylogenies by Edwards and Cavalli-Sforza [4] and to nucleotide sequences by Felsenstein [5]. This computationally intensive but flexible method searches for the tree with highest probability of producing the observed data. The likelihood of each residue in an alignment is calculated based on some model of the substitution process.

Unlike ME and MP, the ML and Bayesian methods make use of all of the information contained within an alignment of DNA sequences. Both ML and Bayesian methods rely on a likelihood function, L(Parameter) = Constant × Prob [Data—Parameter(s)], where the constant is arbitrary and the probability of observing the data conditioned on the parameter is calculated using stochastic models [10]. In ML, the combination of parameters that maximizes the likelihood function is the best estimate. In Bayesian analysis, the joint probability distribution of the parameters is calculated. The posterior probability distribution for the parameters is the likelihood function times the prior probability distribution of the parameters divided by a function of the data. However, unlike ML, Bayesian methods treat parameters as random variables.

Minimum Evolution (ME) is a distance-based approach. In this method, the tree is fit to the data, and the branch lengths are determined using the unweighted least squares method. In this method, distance measures that correct for multiple hits at the same sites are used, and a topology showing the smallest value of the sum of all branches is chosen as an estimate of the correct tree.

When there are a large number of taxa, ME is time consuming, so the neighbor-joining method can be used instead. The Neighbor Joining (NJ) method [17] is a clustering method that minimizes the sum of the branch lengths (this is an approximation to the ME method). The algorithm begins with a star-like structure. Pairwise comparisons are made to determine the most closely related sequences that are connected by a single node, called neighbors. Neighbors form a clade, and the process repeats until the topology is complete.

The NJ and the ME tree are generally the same, but when the number of taxa is small the difference between the trees can be considerable [12]. If a long DNA or amino acid sequence is used, the ME tree is preferable. When the number of nucleotides or amino acids used is relatively small, the NJ method generates the correct topology more often than does the ME method [13, 18]. MEGA uses the close-neighbor-interchange search to examine the neighborhood of the NJ tree to find the potential ME tree.

Unlike NJ, the Unweighted Pair-Group Method with Arithmetic mean (UPGMA) assumes a molecular clock that is constant. This simple distance-based clustering algorithm is significantly less accurate than Neighbor Joining. Each sequence is assigned to its own cluster then new clusters are formed based on having a minimal distance between them. The UPGMA trees are always rooted, and the total branch length from the root to any tip is equal (i.e., the tree is ultrametric). Finding the root requires an outgroup or is given at the midpoint of the longest distance connecting two taxa in the tree.

In this chapter, we will review the mixture tree model and algorithm proposed by Chen and Lindsay [1] in Sect. 7.2 and then in Sect. 7.2.2 present an application to the clustering of the mitochondrial sequences to show that the approach performs well. A simulator that simulates real SNPs sequences with unknown ancestral history will be introduced. Using the simulator we will compare the mixture trees with true trees to evaluate how well the algorithm performs. Comparison with some existing methods including neighbor-joining method, and the maximum parsimony method will also be presented in Sect. 7.3.

7.2 Mixture Tree Algorithm

In this section, we will briefly reviewed the Ancestral mixture model and the Mixture Tree algorithm introduced in the paper Chen and Linsay [1].

7.2.1 Ancestral Mixture Model

The ancestral mixture model implements K-component mutation kernel mixture density to estimate the most common ancestor and the evolving history(phylogeny) of the observed binary DNA sequences. Suppose we observed a sample of binary DNA sequences $\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_n}$ of length L for a fixed mutation rate p. As all the sequences are binary, we can code one state 0 and the opposite 1. If we assume that they evolved from a single ancestor of length L, say $\boldsymbol{\mu}_1$, and we define μ_{1j} as the j^{th} site of $\boldsymbol{\mu}_1$, the mutation kernel density for $\mathbf{X}$ is defined as

$$\kappa(\mathbf{x}|\boldsymbol{\mu}_1, p) = \prod_{j=1}^{L} p^{(x_j-\mu_{1j})^2}(1-p)^{1-(x_j-\mu_{1j})^2} = p^{D(\mathbf{x},\boldsymbol{\mu}_1)}(1-p)^{L-D(\mathbf{x},\boldsymbol{\mu}_1)},$$

where $D(\mathbf{x}, \boldsymbol{\mu}_1) = \sum_{j=1}^{L}(x_j - \mu_{1j})^2$ is the number of disagreements between the site of $\mathbf{x}$ and the corresponding site of $\boldsymbol{\mu}_1$.

If the observed sample is evolving from K different ancestors, say $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$, and we consider ϑ as a random variable with distribution Q, where Q is a discrete distribution with K points of support which are the K ancestors, and $pr(\vartheta = \boldsymbol{\mu}_k) = \pi_k$, where $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$, then we suppose $\mathbf{X}$ is generated by first generating $\vartheta = \boldsymbol{\mu}_k$ from Q, and generating $\mathbf{X} = \mathbf{x}$ from $\kappa(\mathbf{x}|\boldsymbol{\mu}_k, p)$. ϑ is unobserved, and such $\mathbf{X}$ is said to have an ancestral mixture model: $\mathbf{X} \sim A(Q, p)$. The density of $\mathbf{X}$, when Q is discrete, is:

$$f(\mathbf{x}; Q, p) = \sum_{k=1}^{K} \pi_k p^{D(\mathbf{x},\mu_k)} (1-p)^{L-D(\mathbf{x},\mu_k)},$$

which is called a 'Q-mixture of mutation kernels'.

7.2.1.1 Mixture Tree Algorithm

In order to find the MLE of π_j and $\boldsymbol{\mu}_j$, where $j = 1, \ldots, K$, an EM algorithm is employed. Give a value $Q^{(1)} = (\pi_1^{(1)}, \pi_2^{(1)}, \ldots, \pi_{k-1}^{(1)}, \boldsymbol{\mu}_1^{(1)}, \ldots, \boldsymbol{\mu}_K^{(1)})$for the mixture, standard EM calculations give

$$\pi_j^{(t+1)} = \frac{\sum_{i=1}^{n} \delta(j|\mathbf{x_i}; \pi^{(t)}, \boldsymbol{\mu}^{(t)})}{n},$$

where

$$\delta(j|x; \pi^{(t)}, \boldsymbol{\mu}^{(t)}) = \frac{\pi_j \times \kappa(\mathbf{x}|\boldsymbol{\mu}_j)}{\sum_{j=1}^{K} \pi_j \times \kappa(\mathbf{x}|\boldsymbol{\mu}_j)}$$

We then reupdate the δ weights using the new π before update μ. During the E-step, the expected percentage of category 1 occurrences at site s in component j as

$$\upsilon_{js} = \frac{\sum_{i=1}^{n} \delta(j|\mathbf{x_i}; \pi^{(t+1)}, \mu^{(t)}) \times x_{is}}{\sum_{i=1}^{n} \delta(j|\mathbf{x_i}; \pi^{(t+1)}, \mu^{(t)})}$$

and in the M-step, we find the MLE of the parameter by 'voting' according to

$$\hat{\mu}_{js}^{(t+1)} = \begin{cases} 1 & \upsilon_{js} > \frac{1}{2}, \\ 0 & \upsilon_{js} < \frac{1}{2}, \\ either & \upsilon_{js} = \frac{1}{2}. \end{cases}$$

A tie in the third case in this structure of the model is extremely rare and it makes no difference in the EM likelihood.

7.2.1.2 An Alternative Revised Algorithm

The EM algorithm employed in the mixture models has computational problem such as small weight π_i problem. It is nature to propose an alternative revised EM that the weights π_i is not updated. We will call such revised EM the 'FixEM'. Later on, we will compare EM with FixEM in the simulation section.

7.2.2 An Example

In this section we will compare the mixture tree (MT) method with the Neighbor-joining tree and Maximum Parsimony tree in a visual way and give an example of the mixture tree structure by using the real data set in the paper [20]. This dataset can be downloaded from Genbank. There are 530 mtDNA sequences(population) in HVS1 region with different length and they are collected from people living in 17 locations(sub-populations) in East Asia who belong to two official ethnic groups, Miao and Yao, and the sample sizes within each location are different. Before constructing the trees using different methods, we did some necessary manipulations to the sequences:

1. Aligned all the sequences using MEGA4 with default setting.
2. Deleted those sites with gaps
3. Deleted those sites that are not binary
4. When applying mixture algorithm, deleted those sites that are identical

7.2.2.1 Trees Based on the Sample Contains One Random Sequence from Each Sub-population

After applying the above manipulations to all sequences, we constructed trees using four different methods: NJ, MP, ML and MixtureTree algorithm. It is time consuming and resulting tree structure is quite complex if we use all sequences. Therefore, one sequence from each location was randomly chosen and used when constructing trees. Note that the numbers of sequences in the locations are different and some sequences in the location have duplicates, however, sequences from different locations are different. After random selection of one sequence from each location, we have a sample which contains 17 different sequences. Base on the sample, we use MEGA4 to construct the NJ and MP trees which are presented in Figs. 7.1 and 7.2, respectively. Also, we use PHYLIP to construct the ML tree presented in Fig. 7.3. We then deleted all non-binary sites in all sequences then construct the mixture tree. The mixture method uses the frequency of a sequence in the population to assign a weight; here the weights are ones. The mixture tree is constructed and presented in Fig. 7.4.

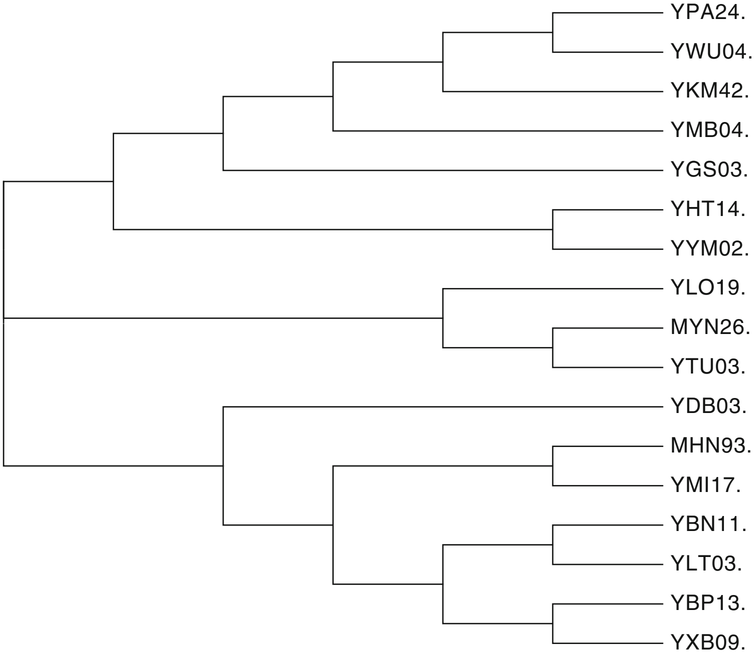


Fig. 7.1 The NJ tree for one sample of the data in Wen et al. [20]

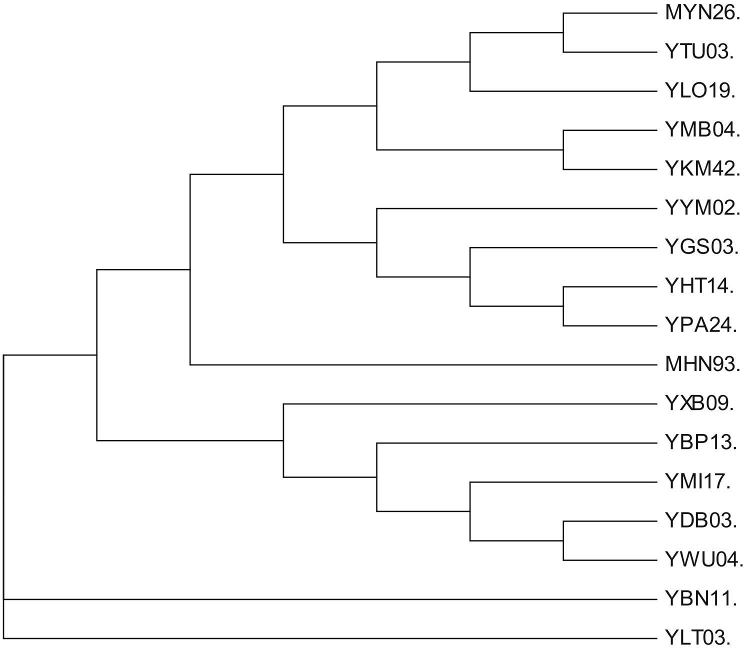


Fig. 7.2 The MP tree for one sample of the data in Wen et al. [20]

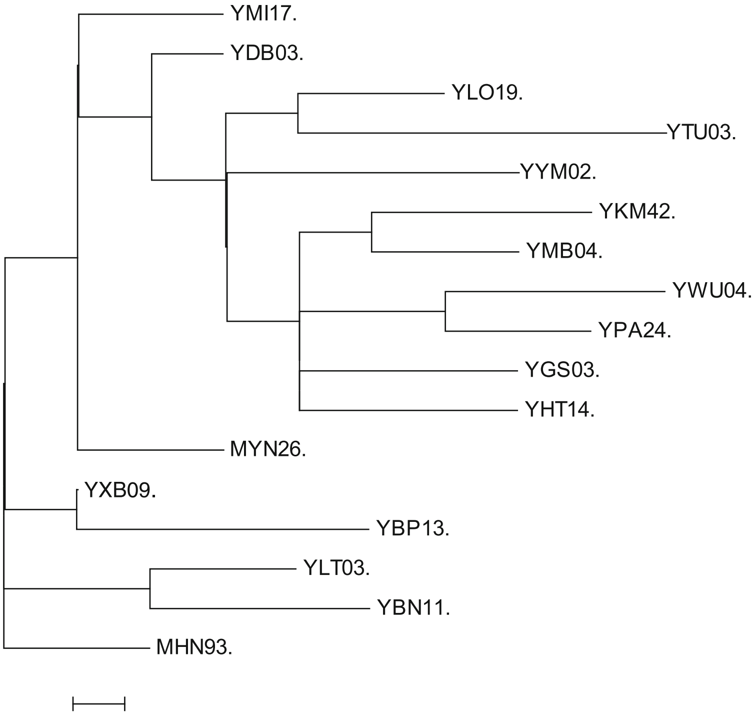


Fig. 7.3 The ML tree for one sample of the data in Wen et al. [20]

7.2.2.2 Trees Based on the Sample Contains all Sequences in the Population

We can also construct trees based on the full set of manipulated sequences in the population by using NJ, MP, ML, and the MT method. The NJ and MP trees can be constructed in MEGA4 and the ML tree can be constructed in PHYLIP. The resulting mixture tree is presented in Fig. 7.4.

7.3 Comparison

7.3.1 Simulator

The simulator we used in comparison of different tree reconstructed methods is ms [9], which is a program to generate samples under a variety of neutral models. A variety of assumptions about migration, recombination rate and population size can be set to generate the designated samples. The samples are generated using the standard coalescent approach in which the random genealogy of the sample is first

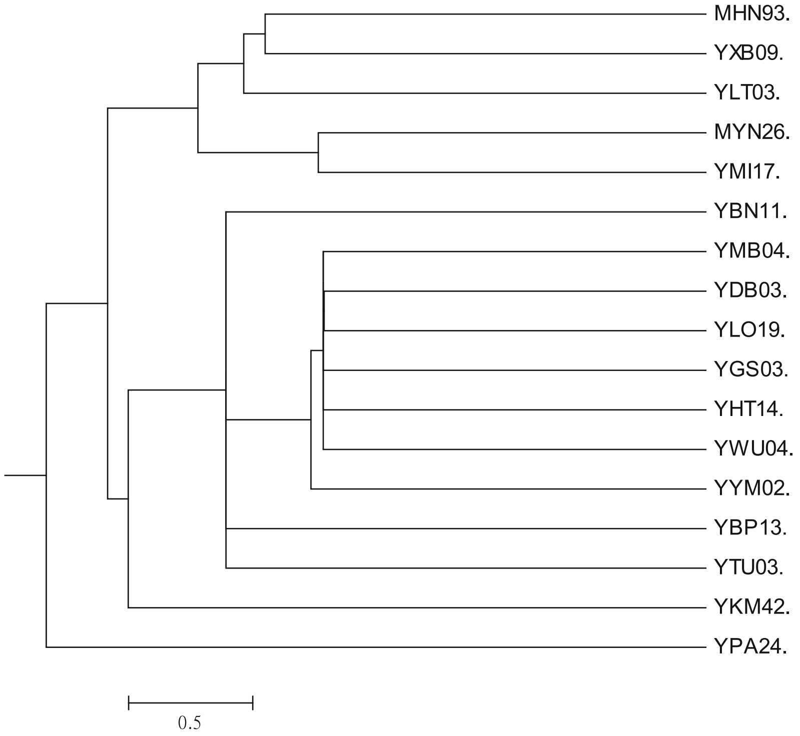


Fig. 7.4 The Mixture Tree for one sample of the data in Wen et al. [20]

generated and then mutations are randomly placed on the genealogy. The simulator can be run under the Unix-Like operating system like Linux.

The basic command line is:

ms nsam nreps -t θ

where

- **nsam** the number of copies of the locus in each sample;
- **nreps** the number of independent samples to generate.
- $\boldsymbol{\theta}$ the mutation parameter, $\theta = (4N_0\mu)$, where N_0 is the diploid population size and where μ is the neutral mutation rate for the entire locus.
- **-t** $\boldsymbol{\theta}$ set value of $4N_0\mu$.

In order to output the gene trees, the option $-T$ needs to be added in basic command. Also, **-s** j needs to be added, if one wants to make samples with fixed number of segregating sites, j.

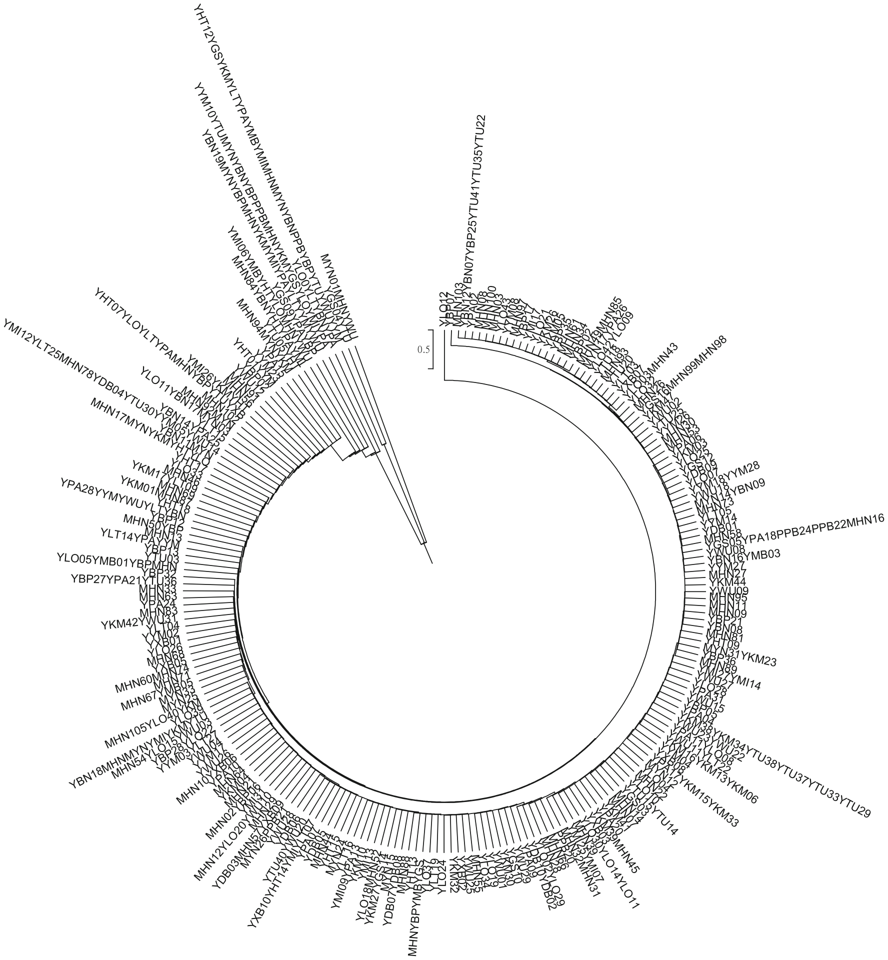


Fig. 7.5 The Mixture Tree for all samples of the data

7.3.2 Comparison

For a set of parameters($\boldsymbol{\theta}$, $\mathbf{s}$), we simulate a sample of size 200 with no identical sequences in each observation and no tie in the corresponding gene tree. Once we have the simulated distinct sequences (suppose it is saved in *tree1.fas*) and no tie in the gene tree (suppose it is saved in *tree1.nwk*), we do the following steps to complete the comparison:

- Change the format of *tree1.fas* to the format which can be used in the mixture tree algorithm and save it as *tree1.txt*;
- Run the mixture tree algorithm with sliding-scale 0.001 using *tree1.txt* and obtain the mixture tree *tree1mm.nwk*;

- Substitute A for 0, G for 1 in *tree1.fas* and reconstruct the Neighbor-joining (*tree1NJ.nwk*) and Maximum Parsimony tree(*tree1MP.nwk*) using MEGA4;
- Using the function *unroot*, *read.tree* and *dist.topo* in the package **ape** in R to compare the distance between *tree1.nwk* and *tree1mm.nwk*, *tree1NJ.nwk*, *tree1MP.nwk*, respectively. Record them.

If there is a tie in the mixture tree, Neighbor-joining tree, and Maximum Parsimony tree during any steps above, we will discard the whole set of sequences.

In order to determine the extent of topological differences between the gene tree(*tree1.nwk*) and the trees created using the other methods (NJ, MP, and MT), Rzhestky and Nei [16] method is implemented. This method is based on the Penny and Hendy's [14] method of sequence partitioning, which provides equivalent numerical values to those obtained using the Robinson and Foulds' [15] method but is simpler to compute. For unrooted bifurcating trees, this distance is twice the number of interior branches at which sequence partitioning is different between the two trees compared. The topological distance can be thought of as the smallest number of transformations required to obtain the simulated tree topology from the tree constructed using the mixture algorithm. The Rzhestky and Nei method is a modification of this distance to take multichotomies into account. These values were standardized by dividing by twice the total number of internal branches. An unrooted bifurcating tree with n haplotypes has $n - 3$ interior branches. Thus, the maximum possible value is $2(n - 3)$. The topological distances were measured and standardized.

7.3.3 Summary of the Analysis

The maximum distance between two trees, given the number of lineage n, using Rzhestky and Nei [16] method, the maximum distance between two trees is $2(n-3)$. So it is reasonable to standardize the distances by dividing each distance by the maximum distance. With different number of different SNPs sequences, the maximum distance between two trees would vary under the Rzhestky and Nei method. The results of the analysis are summarized in Tables 7.1, 7.2, and 7.3.

In the summary Tables 7.1, 7.2, and 7.3, we will call the mixture trees reconstructed via the FixEM algorithm the 'FixMixture', the mixture trees reconstructed via the traditional EM algorithm the 'Mixture'. The 'NJ' means the trees are

Table 7.1 Comparison Results for simulated data with mutation rate 0.0000025 and sample size 200

Mutation rate 0.0000025	Length: 20	No. of Sequences: 5	Samplesize: 200	
	FixMixture	Mixture	NJ	MP
Sum of distance	142	136	156	124
Sum of std. distance	35.5	34	39	31

Table 7.2 Comparison Results for simulated data with mutation rate 0.00000375 and sample size 200

Mutation rate 0.00000375	Length: 10	No. of Sequences: 5	Samplesize: 200	
	FixMixture	Mixture	NJ	MP
Sum of distance	188	168	208	194
Sum of std. distance	47	42	52	48.5

Table 7.3 Comparison Results for simulated data with mutation rate 0.000005 and sample size 200

Mutation rate 0.000005	Length: 10	No. of Sequences: 5	Samplesize: 200	
	FixMixture	Mixture	NJ	MP
Sum of distance	198	146	192	184
Sum of std. distance	49.5	36.5	48	46

reconstructed by the ‘Neighbor-Joining’ algorithm. The ‘MP’ means the trees are reconstructed by the ‘Maximum Parsimony’ algorithm. The ‘Sum of Distance’ is the sum of the Rzhestky and Nei distance of all the units in the sample between mixture tree or Neighbor-joining tree or Maximum Parsimony tree and true gene tree, respectively. The ‘Sum of Std. Distance’ is the sum of Rzhestky and Nei distance of all units in the sample between three different kind trees and gene tree divided by the maximum distance of that unit in the sample, respectively. The ‘Length’ is the length of the simulated sequences in the sample. ‘No. of Sequences’ is the number of sequences in one sample. Please note again that the ‘sum of (Std.)distances’ are the sum of distance between the tree reconstructed by one of these three algorithms and the true gene tree of each unit in the sample. It is obvious that the smaller the distance between two types of trees, the more similar they are. So we can see that the ‘Mixture’ algorithm performed better than at least one algorithm among other tree algorithms in these tables. Sometimes ‘Fix Mixture’ algorithm performed equally better than ‘Mixture’ algorithm, sometimes not. Also, we can see that other two methods are more stable than ‘Mixture’ algorithm and ‘FixMixture’ algorithm, and it is probably due to the fact that ‘Mixture’ and ‘FixMixture’ algorithms embed a more complicated statistical model and take the frequency of each sequence into account when it constructs the tree.

7.4 Discussion

In this chapter we have given an overview of a new method for tree reconstruction called the mixture tree. It provides an estimator of population structure at each point in the past based on a mutational clock. The estimators, unlike some competing methods, are unique. Linking these estimators together over time provides a tree that describes how the population might have evolved. Such a tree can also be used to infer the likely coalescence of lineages, although indirectly. In this chapter we demonstrated how the output of this analysis creates a tree very similar to established methods in the phylogeny literature, and how it can provide a method that is competitive with, but not superior to those competitive methods. In fact, we believe the greater strength of the method lies not in tree construction for distinct phylogenies, but because it provides a clustering method, as well as density estimator, for studies of population structure based on samples from a single population. The theorems are developed in the paper of Lindsay et al. [11] and will be further investigated in the future. Moreover, the current algorithm is based on Bernoulli mixture, which only consider binary sequences. In the future, we will extend it to handle sequences with multiple category and different mutation rates for different types.

Acknowledgements We thank National Science Foundation for the funding – Division of Mathematical Sciences 0714949 to Shu-Chuan Chen and Michael Rosenberg, and National Science Foundation – Division of Mathematical Sciences 0714839 to Bruce Lindsay. In addition, we also acknowledge the support of National Center for Theoretical Sciences (South), Taiwan.

References

1. Chen, S. C., & Lindsay, B. (2006). Building mixture trees from binary sequence data. *Biometrika*, *93*(4), 843–860.
2. Czelsniak, J., Goodman, M., Moncrief, N. D., & Kehoe, S. M. (1990). Maximum parsimony approach to construction of evolutionary trees from aligned homologous sequences. *Methods in Enzymology*, *183*, 601–615.
3. Edwards, A. W. F., & Cavalli-Sforza, L. L. (1963). The reconstruction of evolution. *Annals of Human Genetics*, *27*, 105–106. (also published in Heredity 18:553)
4. Edwards, A. W. F., & Cavalli-Sforza, L. L. (1964). Reconstruction of evolutionary trees. In V. H. Heywood & J. McNeill (Ed.), *Phenetic and phylogenetic classification* (Vol. 6, pp. 67–76). London: Systematics Association Publ.
5. Felsenstein, J. (1981). Evolutionary trees from DNA sequences: A maximum Liklihood Approach. *Journal of Molecular Evolution*, *17*, 368–376.
6. Fisher, R. A. (1912). On an absolute criterion for fitting frequency curves. *Messenger of Mathematics*, *41*, 155–160.
7. Fisher, R. A. (1921). On the "probable error" of a coefficient of correlation deduced from a small sample. *Metron*, *1*, 3–32.
8. Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical transactions of the Royal Society of London A*, *222*, 309–368.
9. Hudson, R. R. (2002). Generating samples under a Wright-Fisher neutral model of genetic variation. *Bioinformatics*, *18*, 337–338.
10. Huelsenbeck, J. P., & Ronquist, F. (2005). Bayesian analysis of molecular evolution using MrBayes. In Nielsen, R. (Ed.), *Statistical methods in molecular evolution*. New York: Springer.
11. Lindsay, B., Markatou, M., Ray, S., Kang, K., & Chen, S. C. (2008). Quadratic distances on probabilities: A unified foundation. *The Annals of Statistics*, *36*(2), 983–1006.
12. Nei, M., & Kumar, S. (2000). *Molecular evolution and phylogenetics*. New York: Oxford University Press.
13. Nei, M., Kumar, S., & Takahashi, K. (1998). The optimization principle in phylogenetic analysis tends to give incorrect topologies when the number of nucleotides or amino acids used is small. *Proceedings of the National Academy of Sciences of the United States of America*, *95*, 12390–12397.
14. Penny, D., & Hendy, M. D. (1985). The use of tree comparison metrics. *Systematic Zoology*, *34*, 75–82.

15. Robinson, D. F., & Foulds, L. R. (1981). Comparison of phylogenetic trees. *Mathematical Biosciences*, *53*, 131–147.
16. Rzhestky, A., & Nei, M. (1992). A simple method for estimating and testing minimum-evolution trees. *Molecular Biology and Evolution*, *9*, 945–967.
17. Saitou, N., & Nei, M. (1987). The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Molecular and Biological Evolution*, *4*, 406–425.
18. Takahashi, K., & Nei, M. (2000). Efficiencies of fast algorithms of phylogenetic inference under the criteria of maximum parsimony, minimum evolution, and maximum likelihood when a large number of sequences are used. *Molecular Biology and Evolution*, *17*, 1251–1258.
19. Tamura, K., Dudley, J., Nei, M., & Kumar, S. (2007). MEGA4: Molecular evolutionary genetics analysis (MEGA) software version 4.0. *Molecular Biology and Evolution*, *24*, 1596–1599.
20. Wen, B., Li, H., Gao, S., et al. (2004). Genetic structure of Hmong-Mien speaking populations in east Asia as revealed by mtDNA lineages. *Molecular and Biological Evolution*, *22*(3), 725–734.

Part II
Expression Data Analysis

Chapter 8
Experimental Designs and ANOVA for Microarray Data

Richard E. Kennedy and Xiangqin Cui

Abstract Microarray experiments are complex, multistep processes that represent a considerable investment of time and resources. Proper experimental design and analysis are critical to the success of a microarray experiment, and must be considered early in the planning of the experiment. Many aspects of experimental design from low-throughput experiments, such as randomization, replication, and blocking, remain applicable to microarray experiments as well. Similarly, the analysis of variance (ANOVA) remains a valid approach for analyzing data from most microarray experiments. However, the high-dimensional nature of microarrays introduces additional considerations into the design and analysis. This chapter provides an overview of the unique statistical challenges presented by microarrays and describes computational methods for implementing these statistical algorithms.

8.1 Experimental Design

Experimental design is defined as the planning of an experiment with the goal of making the experiment more efficient, obtaining the most information with the least expenditure of time, effort, and resources. Proper experimental design is vital to the success of any microarray experiment. Experimental design has long been a subject of study by statisticians, who have developed a considerable literature regarding sound design principles. Much of this literature is devoted to low-throughput experiments such as clinical trials, but the underlying concepts translate well into microarray technologies. This section will briefly review the principles of experimental design as they apply to microarray experiments; the reader is referred to general texts on the topic for a more detailed treatment.

X. Cui (✉) and R.E. Kennedy
University of Alabama-Birmingham, Birmingham, Alabama, USA
e-mail: xcui@ms.soph.uab.edu, rkenned@uab.edu

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_8,

© Springer-Verlag Berlin Heidelberg 2011

8.1.1 Randomization

The first principle of experimental design is randomization, in which the experimental subjects should be randomly assigned to the treatments or conditions to be studied [21]. The purpose of randomization is to eliminate unknown factors that potentially affect results. After a properly performed randomization, the only difference between the experimental groups is the treatment assignment, so that differences in outcome can be attributed to the treatment. The multiple steps in microarray experiments may complicate the process. Besides randomizing samples to the treatments, other factors should also be randomized to avoid bias. For example, the arrays should be randomized in respect to the samples to avoid array order or array batch bias. The order of performing labeling, hybridizations, and scanning may also be randomized to avoid the process timing effect. However, in some cases, it may not be possible to randomize samples to treatments or condition, such as in a comparison of mutant versus wild type mice. In other cases, there are known varying factors present but randomization may not be possible. One such example would be processing batch effect; such an effect occurs but it is not feasible to process all arrays in one batch. In this case, the blocks would be the processing batches and randomization would occur within blocks, as described in later sections.

8.1.2 Replication

Replication is another basic principle of experimental design. The definition of replication is the independent repetition of the same experimental process and/or the independent acquisition of biological observations, so that their error variability can be estimated for evaluating the statistical significance of the observed phenomenon [33]. Such error estimation is essential for applying statistical estimation and inference techniques. Replication makes it possible to estimate the variability associated with the results, which cannot be done with a single occurrence. The estimated variation from replicates permits inference not only about the results that were obtained but also the results that would be obtained if the experiment were repeated in the future.

There is a clear distinction between replications and repeated measurements. To fully understand what a true replicate is requires an understanding of the term *experimental unit*. A true replicate is simply a replicated experimental unit. An experimental unit is defined as the unit that is directly exposed and randomly assigned to the treatment independent of other units [33]. In contrast, repeated measurements refers to taking several measurements from a single occurrence of a phenomenon.

In microarray technology, researchers often use the terms *biological* and *technical* replicates to distinguish replication at different levels [8, 53]. Biological replicates are considered true replicates; while technical replicates are replicates at a lower level than the biological ones, often measurements from the same RNA sample [42, 53]. Because of the complexity of microarray technology, there are potentially many different levels of technical replicates. For example, Fig. 8.1 shows a microarray experiment that has three levels of replication: the first level is at the cell culture level. Four independent cell cultures are established and each receives one of two treatments. Two RNA samples are obtained from each cell culture and each sample is measured with two one-color arrays. In this experiment, the independent cell cultures are biological replicates. The replicates at RNA samples and arrays are technical replicates which are similar to the repeated measurements. They are less useful for identifying significantly expressed genes between the two treatments. However, technical replicates are essential in experiments designed for evaluating the technology and in identifying the sources of variation [56]. The variability between the duplicated arrays estimates the variability of the procedure after RNA extraction, and the variability between the duplicated RNA samples estimates the variability from both RNA extraction and the array hybridization.

As shown in Fig. 8.1, there could be multiple levels of replications in one experiment. Although biological replicates are the most important replication for identifying differentially expressed genes, increasing technical replicates can be a wise choice in certain circumstances. For example, if samples are difficult or expensive to obtain, increased technical replicates will help more accurately measure the gene expression of a relatively small number of samples, especially when the technical variability is larger than the biological variability [10].

8.1.3 Pooling

Pooling is the process of combining several samples into a single sample prior to analysis. Pooling of biological replicates has the potential to reduce biological variability by measuring the average change instead of individual change. Therefore, it is a very appealing process to investigators when biological samples are inexpensive and readily available. Theoretically, pooling can reduce the biological variability to $\frac{1}{n}$, where n is the number of individual biological replicates in each pool. However, this ideal is almost never achieved. One reason is that the pooling process itself has variability [41, 57]. The pool is never a perfect mix of equal amounts for each individual. More importantly, pooling is at the original scale (RNA or tissue level),

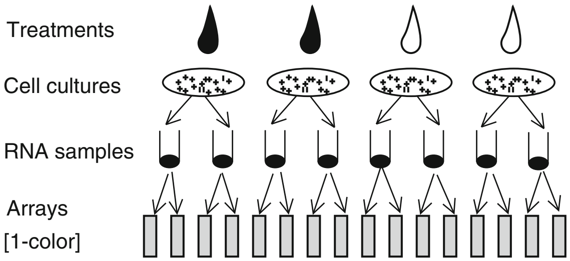


Fig. 8.1 Different levels of replication in a microarray experiment

while variance is often calculated on the log signal intensity level [25, 57]. Despite the limitations of pooling, it can be beneficial in improving the power of the experiment with the same numbers of arrays [27,57]. When the ratio of biological variance to the technical variance is large, the gains from pooling become substantial. Pooling is more beneficial when a small number of arrays and a small number of pools are used for each treatment, although too small a number of pools make the estimate of biological variability unreliable. The decision is not only based on the reduction of the number of arrays but also the cost of individual biological replicates, the ratio of biological and technical variances, and other limits. In certain studies where the amount of RNA extracted from a sample is small, pooling is necessary to obtain enough RNA for one microarray hybridization.

8.1.4 Blocking

Another experimental design principle that is heavily applied to the microarray technology is blocking. Blocking is a way to reduce the effect of variation in factors that are identified but uninteresting. Experimental units are grouped into blocks. It is believed that experimental units within a block are more homogenous than those between different blocks if the blocking is efficient. The treatments or conditions are therefore compared within a block. Blocking originated from field experiments in agriculture where different areas of the field differ in many aspects. To test crop varieties for yield, Fisher [20] divided the field into blocks, within which the field was uniform. Different varieties were then planted randomly within each block. The number of experimental units in each block is called block size. The ideal situation is that the block size is equal to the number of varieties or treatments to be compared, but this is not always achieved. A *complete* block design has the same number of units in each block, which equals the number of treatments to be compared (or treatment combinations in multi-factorial design). All treatment comparisons can be made within each block. Comparisons across blocks provide the information of the between block variabilities. An *incomplete* block design has a block size smaller than the number of treatments (or treatment combinations in multi-factorial design). In this case, not all comparisons can be made within each block. Constructing a balanced incomplete block design is a complicated mathematical problem, although methods for constructing such designs have been described [9, 33, 54, 55].

8.1.4.1 Blocking in Microarray Technology

Block design fits naturally with the two-color microarray platform, where each array is a block of size two for any gene. Investigators have long recognized the variability from spot to spot across arrays due to the differences in DNA quantity, spot morphology, hybridization condition, and scanning settings. Therefore, comparisons are not made between the intensity of one spot to that of another spot for the same gene on a different array. Instead, two paired samples are labeled with different dyes on one array because the two channels of the same spot are more comparable (homogeneous). This feature of the two-color microarray fits into a block design with a block size of two [8, 30, 31]. When there are only two samples to compare, a complete block design results; if there are more than two samples, an incomplete block design results.

The two dyes (Cy3 and Cy5) enable the comparison within a spot. However, each dye has a gene-specific bias, with some genes showing a higher affinity to one dye than the other. When the two samples labeled with different dyes are compared, the higher signal intensity from one dye at a spot may not necessarily mean a higher level of expression of the corresponding gene in the sample labeled by that dye. Rather, the higher signal could come from the higher affinity of that gene for one dye. It has been estimated that about 40% of genes show significant gene-specific dye bias [16]. The gene-specific dye bias cannot be removed through whole chip normalization; it can only be removed by balancing the dyes within treatments when testing for treatment effect. When comparing two samples, the two samples should be labeled in pairs with dye labeling reversed. This pairing strategy is called dye swapping.

8.1.4.2 Reference Designs

For complex experiments, the most intuitive design is the reference design, where all experimental samples are labeled with one dye and compared to a reference sample labeled with the other dye (Fig. 8.2). The ratios of the experimental samples to the reference sample are analysed for treatment or condition effect. Dye bias is eliminated in this design because all experimental samples are labeled with the same dye. The reference sample is often a universal reference or the pool of all other samples. However, it can be individual biological replicates from one (most often the baseline) condition, such as in a time series experiment discussed below. In such cases, there will be different choices for reference samples that fit different experimental goals [45, 53].

8.1.4.3 Loop Designs

Although the use of a universal reference in the reference design corrects for dye bias, it does not contribute to the measurement of the treatment effect, which is often the goal of the experiment. Recognizing this inefficiency, Kerr and Churchill [30] proposed the loop design for comparing multiple samples based on the similarity in blocking structure between microarray and conventional field trials (Fig. 8.2). This design does not involve a reference sample but simply pairs the biological samples on arrays. To make all possible comparisons, the samples are connected in a loop. A graphical representation of such a microarray experiment may be depicted using an arrow to represent an array, with the head pointing to one dye and the tail pointing

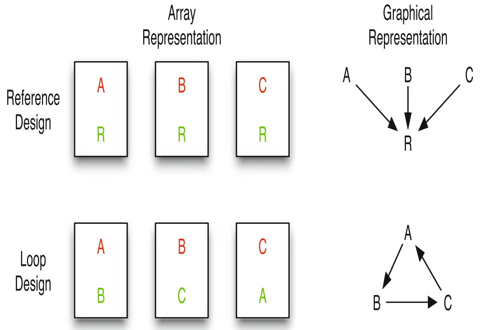


Fig. 8.2 Array and graphical representations of the reference design and loop designs for comparing three samples. In the array representation, the arrays are represented by rectangles. The samples used for hybridization on the array are indicated by letters in the *rectangles*. The dyes are represented by colors of the letters, *green* for the Cy3 dye and *red* for the Cy5 dye. In the graphical representation, array is represented by *arrow* with head pointing to the Cy3 dye and tail pointing to the Cy5 dye. The samples are the nodes of the graphs

to the other. The loop design can balance dyes and achieve a higher efficiency than a reference design when multiple samples are compared. However, construction of a loop design can be more complicated than construction of a reference design if there are different treatments and multiple biological replicates for each treatment. As shown in Fig. 8.3, one strategy is to use one loop for each set of biological replicates when there are equal numbers of biological replicates across treatments [1, 8].

Comparisons between reference and loop designs have shown that both designs have advantages and disadvantages. The reference design is straightforward to implement and extend. The comparison between any two samples in a reference design is within two steps, so the comparisons are equally efficient. The disadvantage of reference design is that half of the measurements are made on the reference sample, so it is not very efficient in the use of arrays. In contrast, the overall efficiency of a loop design is higher than that of a reference design [1, 8, 29, 30, 48, 53]. For example, for the two designs shown in Fig. 8.2, the average variance of the reference design is 2, but it is only 0.67 for the loop design [53]. However, the efficiency of some comparisons can be low when the loop becomes larger [14, 30]. In addition, it is rather complicated to construct a loop design to achieve optimal efficiency when there are multiple samples [1, 30, 48]. It is not obvious how to extend a loop design, although proposed methods for doing so have been described [1]. Finally, the loop design is less robust to missing or poor quality chips [2]. If one of the chips in the loop is lost or of poor quality, comparisons among the remaining chips in the loop can still be made, but the measurement error increases significantly. In contrast, the remaining comparisons in the reference design are unaffected by the loss of a single chip, unless the reference itself is defective [7].

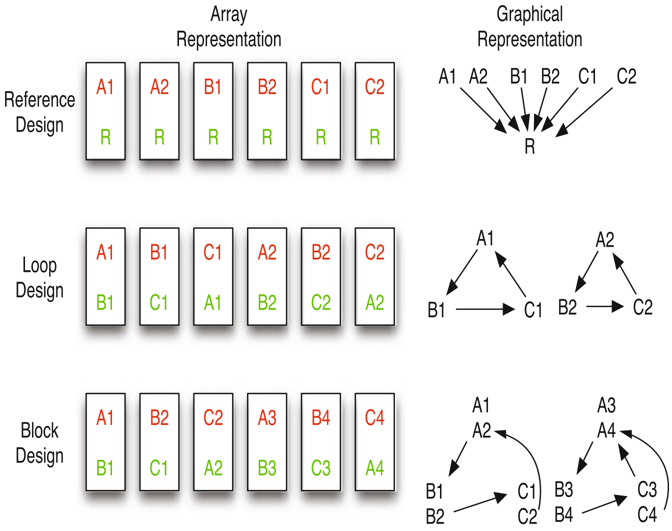


Fig. 8.3 Array and graphical representations of designs with biological replicates. The three treatments are represented by three letters. Biological replicates within treatments are represented by the combination of letters and numbers. In the array presentation, arrays are represented by *rectangles* and dyes are represented by colors, *green* for the Cy3 dye and *red* for the Cy5 dye. In the graphical representation, arrays are represented by *arrows* with head pointing to the Cy3 dye and tail pointing to the Cy5 dye. For the same number of arrays, the balanced block design can accommodate four biological replicates, while the other two designs can only accommodate two biological replicates

8.1.4.4 Balanced Incomplete Block (BIB) Design

Another design for complex experiments is the balanced incomplete block design, where each biological replicate is labeled once and samples from different treatments are paired on arrays with dyes balanced in respect to treatments (Fig. 8.3). Similar to the loop design, a reference sample is not used in the BIB design. The difference between a balanced block design and loop design is how many times a biological sample is labeled. In a block design, each sample is labeled once, and the balance of dyes is achieved at the treatment level. In a loop design, each sample is labeled twice by both dyes, and the balance of dyes is achieved at the sample level. Because balanced block design can incorporate more biological samples without using technical replicates for the same number of arrays, it can be more efficient in testing for the treatment effect than a loop design. However, block design cannot be used for classifying individual samples because the cross array comparison is often not possible due to the large variation [14, 15]. Loop designs and reference designs can be used for classifying individual samples.

8.1.4.5 Other Potential Blocking Factors

Besides the pairing of samples on two-color microarrays, there are other applications of blocking in microarray experiments. For example, if the arrays in an experiment come from different lots, the array lot may be treated as a blocking factor by hybridizing an equal number of samples from each treatment within each lot. Similarly, if the number of arrays is too large to process in a single batch, the batch can be treated as a block by processing the same number of arrays from each treatment in each batch. If two technicians are working on the same experiment, the technician may be treated as a blocking factor, with each person processing a full set of samples from each treatment. In these cases, the variation of array lots, processing batches, and technicians will be blocked out and the results will be less biased.

8.1.5 Row-Column Designs

Although block designs are commonly used in microarray studies, the alternative row-column design has also been proposed. Block designs are intended to have only one source of variation besides treatment, i.e., the blocks. Row-column designs are intended to have two sources of variation, i.e., the rows and columns. Modeling two sources of variation instead of one can reduce the experimental error in analyses. In the microarray setting, the row-column design may naturally be applied to two-color arrays, with rows representing dyes and the columns slides. The most familiar example of the row-column design is the Latin Square, in which the number of rows and columns both equal the number of treatments. The row-column design may be reduced to a block design by ignoring the rows (dyes) and considering only the columns (slides) as blocks. The row-column design may be more efficient than block designs, particularly for larger studies [2, 7], but at the cost of increased complexity. To offset this, some authors have developed published tables for moderately sized experiments, and provided design principles for larger studies [2, 7, 37]. The analysis of studies with row-column designs is also more complicated than analysis of block designs, although both can be accomplished using currently available software packages.

8.1.6 Experimental Design for Classification Studies

Microarray technology is commonly used in the clustering and classification studies, especially in the classification of tumors [39, 42]. A clustering study is often used to examine the gene expression level across individual samples and to assess similarities or dissimilarities across genes and across individuals. Classification studies are used to predict the classes of samples based on the expression of a subset of genes that are selected from the training set of samples. The experimental designs for these applications are different from those used for identifying differentially expressed genes across treatments, conditions, or classes. In clustering and classification studies, there are often no biological replicates, and individual samples are the main interest. In addition, there are often many more individual samples to be compared.

The design for clustering and classification is often very simple when one-color microarray is used, with one array for each sample. For two-color microarrays, the pairing scheme must be considered. Unlike experiments to compare classes or treatments, the application of loop and block designs is limited in clustering and classification experiments [15]. The reference design is the primary choice due to its practical advantage and extendibility. The loop design can be used, but may be complicated and inefficient. The balanced incomplete block design is not applicable due to the lack of dye balance and confounding of individual effects with array effects.

8.1.7 Experimental Design for Time Course Experiments

Time course experiments, which profile gene expression at different times or developmental stages, are used to reveal the dynamics of gene expression [4]. The design of a time series microarray experiment has some similarity to the design of experiments for comparing different classes of samples. The reference design, loop design, and direct comparison are the building blocks of designs for a time series experiment. However, the comparisons of interest have a greater role in the selection of designs for a time course experiment. If the comparison of interest is the consecutive time point in the series, direct comparison of neighboring points will be the most efficient use of the arrays [53]. On the other hand, if the comparison of the initial time point to all other time points is the most important comparison, direct comparison between the initial time point and the remaining points is beneficial. This design is very similar to a reference design, except that the reference sample is of interest and biological replicates are desired for this reference sample. In addition, dye balance needs to be considered between the reference and other samples. This becomes an alternative reference design [45]. If all comparisons are of interest, the alternative reference design, the interwoven loop design, and the combination of both (carriage wheel design) are some choices [32]. The alternative reference design has the advantage of equal efficiency for any comparisons between the baseline point and the other points, or among the other points, although it may be less efficient than other more direct comparisons in overall efficiency. The interwoven loop design uses multiple loops connecting the samples to avoid time points that are too far away in one loop. The carriage wheel design is especially suitable for experiments that are intended to compare not only adjacent time points but also the initial point and all other points. This design uses the initial point as the reference and connects the rest of the time points consecutively into a loop [32].

8.1.8 Experimental Design for eQTL Studies

A more recent application of microarray is in the mapping of quantitative trait loci (QTL). The conventional analysis of QTL uses quantitative phenotypic traits, such as blood pressure or body weight, which are measured from each individual in a genetically segregating family of individuals in pedigrees. In recent years, microarrays have been used to profile the gene expression of each individual in a QTL mapping population [13, 23]. The expression of each gene is then treated as a quantitative trait for QTL mapping. The design of this type of experiment is complex, with design issues related to the QTL mapping aspect and design issues related to the microarray aspect. For the microarray aspect, one array is simply used for each individual for a one-color microarray platform. For a two-color microarray platform, the pairing of samples on each array must be considered. Although the type of QTL population can vary from study to study, such as model organism back cross, recombinant inbred lines, or family pedigrees, the principles for pairing the samples on arrays are the same. The first thing to be clear about is the objective of the experiment and to identify the main interest effects to be mapped. The second is to pair the most dissimilar samples regarding the interested effect on the same array in a block design. The most dissimilar samples can often be identified based on the marker genotypes. The reference and loop designs are also applicable, but are less efficient for mapping genetic factors controlling the expression of each gene [6, 22].

8.1.9 Experimental Design for ChIP-Chip Studies

Chromatin immunoprecipitation (ChIP) is used to identify protein binding sites on chromosomal DNA. A crosslinking agent, such as formaldehyde, is used to immobilize DNA-binding proteins to their active site on chromatin. After the DNA is sheared, the target DNA-protein fragments are selected using techniques such as immunoprecipitation or affinity purification. The crosslinking is then reversed, allowing the bound DNA fragments to be analyzed. This process enriches the DNA fragments for regions to which the target protein are bound. Recently, microarrays have been used to perform ChIP on a genome-wide scale (ChIP-chip). Rather than analyzing only a few DNA regions for enrichment using ChIP, intensity measurements from a ChIP-chip experiment examine the enrichment of thousands of regions simultaneously. These types of studies have specific requirements for both the selection of microarrays and the design of the experiment [5]. Expression microarrays, which target genic regions of the DNA, are often unsuitable for measuring protein binding, which typically occurs in intergenic regions. Similarly, cDNA probes may be problematic as the intergenic regions may be spliced out of the transcript. Thus, the microarray platform for ChIP-chip experiments is often designed specifically to target promoter regions of genes, or similar regions of DNA-protein interaction. An alternative is the *tiling* array, which has probes for sequences regularly spaced along the chromosome, rather than targeting specific genes. Another consideration

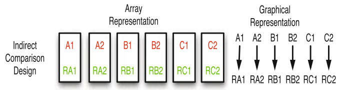


Fig. 8.4 Array and graphical representations of ChIP-chip experiment design. The three treatments are represented by three letters. Biological replicates within treatments are represented by the combination of letters and numbers. In the array presentation, arrays are represented by *rectangles* and dyes are represented by colors, *green* for the Cy3 dye and *red* for the Cy5 dye. In the graphical representation, arrays are represented by *arrows* with head pointing to the Cy3 dye and tail pointing to the Cy5 dye

in the design of ChIP-chip studies is the selection of the hybridization reference and the control experiment. The hybridization reference controls for nonspecific binding and enrichment in the analysis of signals from the DNA fragments. Usually sheared genomic DNA from the experimental organism is the choice for a hybridization reference, and comparison of the intensities between the sample and the hybridization reference tests are used to detect regions of enrichment due to DNA-protein binding. The control experiment is intended to correct for variation due to sources other than protein binding to a DNA region, such as nonspecific antibody interactions. Most experiments use a mock immunoprecipitate (IP), in which the ChIP experiment is performed with the omission of the antibody or the substitution of a nonspecific antibody. Thus the typical design for ChIP-chip experiments, depicted in Fig. 8.4, involves indirect comparisons among experimental conditions. Finally, dye swaps to control for dye bias are uncommon in ChIP-chip studies; fortunately, preliminary evidence indicates that the effect of dye bias on this type of experiment is small [38], though this needs further confirmation.

8.2 Analysis of Variance

8.2.1 The General Linear Model

The earliest and simplest approach for determining differential expression in microarray experiments was to examine the fold change, which is the ratio of the intensity values for a gene in the two conditions of interest. However, investigators quickly realized that fold change was an inadequate measure, as it does not account for the variability of expression measurements [52]. Statistical tests of hypotheses are necessary to assess the reliability of findings from microarray experiments [19]. Many different types of analyses are suitable for microarray data, but variants of the *general linear model* are among the most widely used, particularly for assessing differential expression [24]. The general linear model is not specific to microarrays, but is a nonspecific methodology that serves as the "workhorse" of most statistical analyses. The formulation of the general linear model assumes that the measured or *dependent* variable can be explained by a set of predictor or *independent* variables plus random (measurement) error. The general linear model further assumes this relationship is *linear*; that is, the relationship between the measured and predictor variables is a straight line if there is only one predictor, or the equivalent of a straight line if there are multiple predictors. Finally, the random or measurement error is assumed to follow the familiar *normal distribution*. These assumptions greatly simplify the mathematics and computational algorithms for analysis of the general linear model. In the context of microarrays, the independent variables would be the intensity measurement for the gene of interest, while the predictor variables would be a set of explanatory variables such as age, race, or disease status. The typical microarray experiment involves the measurement of multiple genes simultaneously, which may be manipulated mathematically as a *vector*, or group of related numbers. Standard statistical texts on linear models provide more information on vector algebra and the computational details of analyzing the general linear model, which are beyond the scope of this chapter.

The most common implementation of the general linear model in microarray experiments is the analysis of variance (ANOVA), which compares gene intensities among multiple *classes* or groups. In this case, the predictor variables denote class membership (such as normal versus diseased, or treated versus untreated) of each gene on each chip. The predictor variables in the ANOVA model are also called *factors* or *effects*. The general linear model also subsumes the familiar t-test, which limits comparisons to two classes and thus represents a more specific case of the ANOVA model. The general linear model subsumes linear regression as well, which examines the relationship between gene intensities and quantitative measures. In linear regression, the predictor variables do not denote categories of class membership, but measurements of numerical quantities such as age, height, or weight. Regression models for microarray data have been developed [40], but are not commonly used and will not be dealt with further in this chapter. There is a considerable volume of literature on the use of general linear models in the analysis of experiments, and the reader is referred to standard texts on the subject for more detailed coverage. However, the nature of microarray experiments raises specific analytical issues, which will be considered in the remainder of this section.

8.2.2 Fixed Versus Random Effects

The factors, or effects, in the ANOVA model may be broadly classified as *fixed* or *random*. For fixed effects, all levels of the effect are assumed to be enumerated in the experiment, and repetition of the experiment would result in the same levels being present. Examples would include gender (male and female) and dye labels (red and green), which do not change from experiment to experiment. In models incorporating only fixed effects, all of the variation is assumed to be due to random or measurement error. For random effects, the levels in the experiment represent a random sample from a population of possible levels. Thus, not all levels are present in the experiment, and repetition of the experiment would result in different levels being used. Arrays would be a primary example, as the arrays used in an experiment represent a random sample from the larger population of arrays that could have been used. While fixed effects models have only a single source of variation due to measurement error, the random selection of levels in a random effects model introduces an additional source of variation into the model. Analysis of random effects models incorporates this additional source of variation by requiring that the distribution of the random factors be specified, with the normal distribution commonly used. Because the random effects are assumed to be drawn from a larger population of possible effects, the inferences from a random effects model may be more appropriately generalized than inferences from a fixed effects model. For example, if array is analyzed as a fixed effect, the implicit assumption is that there is a single array effect that is constant across experiments. Such an assumption may be implausible based on array technology, where the manufacturing process may introduce differences among batches of arrays. This would mean that the inferences from the analysis can only properly be applied to arrays having the same array effect, and not to arrays having different characteristics. However, if array is analyzed as a random effect, the assumption is that the array effect varies across experiments. Because the random effects model accounts for the variation due to these differences, the inferences from the analysis can be applied not only to the arrays used in the experiment but also to arrays with different characteristics used in other experiments. In general, effects such as arrays should be analyzed as random effects for this reason. The use of a random effects model also allows for the factors to be correlated with each other, which is not possible with a fixed effects model. For example, the effects of a drug administered at different time points in an experiment, which would likely be correlated, can be captured using a random effects model. An ANOVA model that has both fixed and random effects is called a *mixed model*. The designation of factors as fixed or random is not always straightforward and can have substantial influence on the results obtained.

8.2.3 Error Specification

The handling of random or measurement error requires special consideration when using ANOVA for microarray data. A standard ANOVA model assumes that the errors are homogenous. In terms of a microarray experiment, this would mean that the variation due to measurement error is identical across probesets, which is implausible biologically as the amount of variation between genes is usually considerable [49]. Furthermore, when using a single global variance, rankings based on the p-values of the t or F tests under the general linear model reduce to rankings based on the fold change [11], which has already been noted to be problematic. An alternative would be to allow errors to be completely heterogenous, so that each gene has its own random or measurement error. This would mean that the variation due to measurement error is unique for each gene, which would be biologically justifiable. However, this also means information about the variation for a gene can only be obtained from measurements of that particular gene; in contrast, information about variation for the homogenous variance model can be obtained from measurements of all genes, since each is assumed to have identical variation. This leads to the former having very low statistical power to detect significant differences compared to the latter, which makes the heterogenous variance approach unsuitable for analyses. Thus, the most common approach is to combine the two into a "moderated" variance estimate. This allows the variance to differ by gene, but borrows information across genes to increase statistical power.

A number of approaches have been proposed for borrowing information across genes. A common approach is to estimate the variation using Bayes or empirical Bayes procedures, which allow one to impose specific relationships among the variances [3, 17, 26, 34, 36, 43]. This is implemented in the *limma* package available from Bioconductor [44]. Non-Bayesian methods for moderating the variance have also been developed. The t-test in the SAM software (http://www-stat.stanford.edu/~tibs/SAM/) adds a small bias constant to the gene-specific variances to attenuate the effect of small variances [47]. Although this does moderate the variances, it fails to utilize information across genes. Other researchers suggest that the gene-specific variances be considered as a sample from a common distribution, which shares similarities to the Bayesian approach but leads to different estimators [49]. Finally, Cui and colleagues [12] describe a shrinkage estimator for the gene-specific variances based on the James-Stein estimator.

8.2.4 Computational Issues

There are several computational issues in the analysis of the general linear model. Most of these are not specific to microarray and have been described in detail elsewhere [28, 35]. However, one topic of special interest in microarray experiments is the calculation of test statistics and the computation of p-values. As all measurements are subject to error, the purpose of a statistical analysis is to quantify the variation in an experiment, and to determine if this variation may be plausibly attributed to the factor of interest (such as drug treatment) or to random error. In the ANOVA model, this is done by comparing the variation between groups to the variation within a group; the latter is assumed to be due to random error, while the former is due to the factor of interest. If the variation between groups is similar to the variation within groups, this implies that the factor of interest does not contribute significantly to the observed variation in the experiment, while variation between groups exceeding variation within groups implies that the factor does contribute significantly. A formal comparison of the between-group and within-group variation uses a test statistic, which is based on the ratio of the two types of variation; most analyses in ANOVA use the familiar F statistic. When using a mixed effects model with multiple sources of variation, the formulation of the test statistic may become complicated in selecting the appropriate variation to use in the computation, but the principle of testing remains the same.

After a test statistic is computed, it is convenient to convert it to a p-value. Genes with p-values falling below a prescribed level may be regarded as significant. Reporting p-values as a measure of evidence allows some flexibility in the interpretation of a statistical test by providing more information that a simple dichotomy of significant or not significant at a predefined level. Traditionally, p-values are found by reference to a statistical distribution table or by use of mathematical formulas. Both methods rely on the assumption that the test statistic follows a particular distribution (such as the F distribution), so that the test statistic can be compared to the expected value based on the mathematical description of the distribution. Thus, if the assumption that the test statistic follows a particular distribution is incorrect, the inferences made using that test statistic may also be incorrect. Furthermore, if a test statistic cannot be shown to follow a particular distribution, calculation of the p-value using mathematical formulas may be impossible.

8.2.4.1 Permutation Analysis

Permutation analysis does not require the assumption that the test statistic follows a particular distribution, but the experiment should be large enough that a sufficient number of permutations can be obtained. Permutation analysis relies on the assumption that, under the null hypothesis, all conditions would be expected to be equal. In a microarray experiment, this would mean that, under the null hypothesis of no differential expression, a gene should have similar intensity values on all of the arrays being examined. Because of this principle of *exchangeability*, the samples may be shuffled between the groups or conditions of interest. If the null hypothesis is true, then shuffling the samples should not significantly change the test statistic; if the test statistic does change significantly, this is evidence that the null hypothesis is false. In a microarray experiment, the samples would be shuffled between groups (such as treated versus untreated). If no differential expression has occurred, the test statistic should not be significantly altered by this shuffling; a significant change in the test statistic would be evidence of differential expression. By repeatedly shuffling over all possible combinations of samples, any changes in the test statistic can be quantified to determine if the change is greater than what would be expected from random variation. Permutation analysis requires that sufficient samples be available for constructing the combinations; if too few samples are available, shuffling the samples does not give an adequate picture of the possible changes in the test statistic. A minimum of about 6 replicates per condition (yielding a total of 924 distinct permutations) is recommended for a two-sample comparison. Pooling the test statistics of all genes in the permutation analysis has been proposed to overcome the limitation of small sample sizes [46]. However, this approach assumes that the distribution of the null statistics of differentially and non-differentially expressed genes is the same. The variability of the permuted test statistics is increased when differentially expressed genes are included in the permutation, which may lead to conservative p-values that miss true positives [50, 51]. Permutation over a subset of genes, which are intended to contain only non-differentially expressed genes, would reduce the conservativeness of the p-values. The differentially and non-differentially expressed genes are generally not known, and such a classification is often the goal of a microarray experiment. Thus, various methods have been proposed to identify the gene subset for permutation analysis [18, 50, 51].

8.2.4.2 Bootstrapping

For large experiments, bootstrapping may be used instead of permutation analysis. While the latter examines all possible combinations of samples, bootstrapping selects a random set with replacement from the samples, and the test statistic is then calculated using the randomly selected set. By repeatedly resampling sets from the sample and computing the test statistic, an approximate distribution of the test statistic can be obtained. A p-value may then be calculated as the proportion of resampled test statistics from the bootstrap that exceed the test statistic obtained from the experiment. As with permutation analysis, bootstrapping requires sufficient samples for performing the resampling procedure. While permutation analysis examines all possible combinations of samples, bootstrapping examines only a resampled set of observations from the samples. Thus, bootstrapping also must have sufficient repetitions of the resampling to give an adequate picture of changes to the test statistic.

8.2.4.3 Limitations

Both permutation analysis and bootstrapping do not require the assumption that the test statistic follow a particular *theoretical* distribution, instead basing tests of significance on the *empirical* distribution obtained from shuffling and resampling. However, both require that the principle of exchangeability be met, which assumes there are no differences except for random error between samples under the null hypothesis. This assumption is frequently overlooked in analyses and can lead to erroneous conclusions. Finally, both permutation analysis and bootstrapping are computationally intensive because of the large number of combinations or resampling that must be constructed and individually analyzed. This may make these procedures unsuitable in experiments where the individual analysis is time-consuming.

8.3 Conclusions

Microarray experiments are complex, multistep processes that represent a considerable investment of time and resources. Many aspects of experimental design and analysis from low-throughput experiments, such as clinical trials, remain applicable to microarray experiments as well. However, the high-dimensional nature of microarrays introduces additional considerations into the design and analysis.

Proper experimental design and analysis are critical to the success of a microarray experiment, and must be considered early in the planning of the experiment.

References

1. Altman, N. S., & Hua, J. (2006). Extending the loop design for two-channel microarray experiments. *Genetical Research*, *88*, 153–163.
2. Bailey, R. A. (2007). Designs for two-colour microarray experiments. *Applied Statistics*, *56*(4), 365–394.
3. Baldi, P., & Long, A. D. (2001). A Bayesian framework for the analysis of microarray expression data: Regularized t-test and statistical inferences of gene changes. *Bioinformatics*, *17*(6), 509–519.
4. Bar-Joseph, Z. (2004). Analyzing time series gene expression data. *Bioinformatics*, *20*, 2493–2503.
5. Buck, M. J., & Lieb, J. D. (2004). Chip-chip: Considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics*, *83*, 349–360.
6. Bueno Filho, J. S., Gilmour, S. G., & Rosa, G. J. (2006). Design of microarray experiments for genetical genomics studies. *Genetics*, *174*, 945–957.
7. Chai F.-S., Liao C.-T., & Tsai S.-F. (2007). Statistical designs for two-color spotted microarray experiments. *Biometrical Journal*, *49*(2), 259–271.
8. Churchill, G. A. (2002). Fundamentals of experimental design for cDNA microarrays. *Nature Genetics*, *32* Suppl, 490–495.
9. Cox, D. R. (1958). *Planning of experiments*. New York: Wiley.
10. Cui, X., & Churchill, G. A. (2003). How many mice and how many arrays? replication of cDNA microarray experiments. In M. L. Simon & T. A. Emily (Eds.), *Methods of microarray data analysis III*. New York: Kluwer Academic Publishers.
11. Cui, X., & Churchill, G. A. (2003). Statistical tests for differential expression in cDNA microarray experiments. *Genome Biology*, *4*(4), 210.
12. Cui, X., Hwang, J. T., Qiu, J., Blades, N. J., & Churchill, G. A. (2005). Improved statistical tests for differential gene expression by shrinking variance components estimates. *Biostatistics*, *6*(1), 59–75.
13. de Koning, D. J., & Haley, C. S. (2005). Genetical genomics in humans and model organisms. *Trends in Genetics*, *21*, 377–381.
14. Dobbin, K., Shih, J. H., & Simon, R. (2003). Statistical design of reverse dye microarrays. *Bioinformatics*, *19*(7), 803–810.
15. Dobbin, K., & Simon, R. (2002). Comparison of microarray designs for class comparison and class discovery. *Bioinformatics*, *18*, 1438–1445.
16. Dobbin, K. K., Kawasaki, E. S., Petersen, D. W., & Simon, R. M. (2005). Characterizing dye bias in microarray experiments. *Bioinformatics*, *21*, 2430–2437.
17. Efron, B., Tibshirani, R., Storey, J. D., & Tusher, V. (2001). Empirical Bayes analysis of a microarray experiment. *Journal of American Statistical Association*, *96*(456), 1151–1160.
18. Fan, J., Chen, Y., Chan, H. M., Tam, P. K. H., & Ren, Y. (2005). Removing intensity effects and identifying significant genes for affymetrix arrays in macrophage migration inhibitory factor-suppressed neuroblastoma cells. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(49), 17,751–17,756.
19. Firestein, G. S., & Pisetsky, D. S. (2002). DNA microarrays: Boundless technology or bound by technology? Guidelines for studies using microarray technology. *Arthritis & Rheumatism*, *46*(4), 859–861.
20. Fisher, R. A. (1926). The arrangement of field experiments. *Journal of the Ministry of Agriculture of Great Britain*, *33*, 503–513.

21. Fisher, R. A. (1935). *The design of experiments*. Edinburgh: Oliver and Boyd.
22. Fu, J., & Jansen, R. C. (2006). Optimal design and analysis of genetic studies on gene expression. *Genetics*, *172*, 1993–1999.
23. Gibson, G., & Weir, B. (2005). The quantitative genetics of transcription. *Genetics*, *21*, 616–623.
24. Jafari, P., & Azuaje, F. (2006). An assessment of recently published gene expression data analyses: Reporting experimental design and statistical factors. *BMC Medical Informatics and Decision Making*, *6*, 27.
25. Kendziorski, C. M., Irizarry, R. A., Chen, K. S., Haag, J. D., & Gould, M. N. (2005). On the utility of pooling biological samples in microarray experiments. *Proceedings of the National Academy of Sciences of the United of States America*, *102*, 4252–4257.
26. Kendziorski, C. M., Newton, M. A., Lan, H., & Gould, M. N. (2003). On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles. *Statistics in Medicine*, *22*(24), 3899–3914.
27. Kendziorski, C. M., Zhang, Y., Lan, H., & Attie, A. D. (2003). The efficiency of pooling mRNA in microarray experiments. *Biostatistics*, *4*, 465–477.
28. Kennedy, W. J, & Gentle, J. E. (1980). *Statistical computing*. New York: Marcel Dekker.
29. Kerr, M. K. (2003). Design considerations for efficient and effective microarray studies. *Biometrics*, *59*, 822–828.
30. Kerr, M. K., & Churchill, G. A. (2001). Experimental design for gene expression microarrays. *Biostatistics*, *2*, 183–201.
31. Kerr, M. K., & Churchill, G. A. (2001). Statistical design and the analysis of gene expression microarray data. *Genetical Research*, *77*(2), 123–128.
32. Khanin, R., & Wit, E. (2005). Design of large time-course microarray experiments with two channels. *Applied Bioinformatics*, *4*, 253–261.
33. Kuehl, R. O. (2000). *Design of experiments: Statistical principles of research design and analysis*. New York: Duxbury Press.
34. Lonnstedt, I., & Speed, T. (2002). Replicated microarray data. *Statistica Sinica*, *12*, 31–46.
35. Monahan, J. F. (2001). *Numerical methods of statistics*. New York: Cambridge.
36. Newton, M. A., Kendziorski, C. M., Richmond, C. S., Blattner, F. R., & Tsui, K. W. (2001). On differential variability of expression ratios: Improving statistical inference about gene expression changes from microarray data. *Journal of Computational Biology*, *8*(1), 37–52.
37. Nguyen N.-K., & Williams, E. R. (2006). Experimental designs for 2-colour cDNA microarray experiments. *Applied Stochastic Models in Business and Industry*, *22*, 631–638.
38. Ponzielli, R., Boutros, P. C., Katz, S., Stojanova, A., Hanley, A. P., Khosravi, F., Bros, C., Jurisica, I., & Penn, L. Z. (2008). Optimization of experimental design parameters for high–throughput chromatin immunoprecipitation studies. *Nucleic Acids Research*, *36*, e144.
39. Quackenbush, J. (2006). Microarray analysis and tumor classification. *New England Journal of Medicine*, *354*, 2463–2472.
40. Segal, M. R., Dahlquist, K. D., & Conklin, B. R. (2003). Regression approaches for microarray data analysis. *Journal of Computational Biology*, *10*, 961–980.
41. Shih, J. H., Michalowska, A. M., Dobbin, K., Ye, Y., Qiu, T. H., & Green, J. E. (2004). Effects of pooling mRNA in microarray class comparisons. *Bioinformatics*, *20*, 3318–3325.
42. Simon, R. (2003). Diagnostic and prognostic prediction using gene expression profiles in high-dimensional microarray data. *British Journal of Cancer*, *89*, 1599–1604.
43. Smyth, G. K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, *3*, Article3.
44. Smyth, G. K. (2005). Limma: Linear models for microarray data. In R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, & W. Huber (Eds.), *Bioinformatics and computational biology solutions using R and bioconductor* (pp. 397–420). New York: Springer.
45. Steibel, J. P., & Rosa, G. J. (2005). On reference designs for microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, *4*, Article36.

46. Storey, J. D., & Tibshirani, R. (2003). SAM thresholding and false discovery rates for detecting differential gene expression in DNA microarrays. In G. Parmigiani, E. S. Garret, R. Irizarry, & S. Zeger (Eds.), *The analysis of gene expression data: An overview of methods and software* (pp. 272–290). New York: Springer.
47. Tusher, V. G., Tibshirani, R., & Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America*, *98*, 5116–5121.
48. Wit, E., Nobile, A., & Khanin, R. (2005). Near-optimal designs for dual channel microarray studies. *Applied Statistics*, *54*, 817–830.
49. Wright, G. W., & Simon, R. M. (2003). A random variance model for detection of differential gene expression in small microarray experiments. *Bioinformatics*, *19*(18), 2448–2455.
50. Xie, Y., Pan, W., & Khodursky, A. B. (2005). A note on using permutation-based false discovery rate estimates to compare different analysis methods for microarray data. *Bioinformatics*, *21*(23), 4280–4288.
51. Yang, H., & Churchill, G. A. (2007). Estimating p-values in small microarray experiments. *Bioinformatics*, *23*(1), 38–43.
52. Yang, I. V., Chen, E., Hasseman, J. P., Liang, W., Frank, B. C., Wang, S., Sharov, V., Saeed, A. I., White, J., Li, J., Lee, N. H., Yeatman, T. J., & Quackenbush, J. (2002). Within the fold: Assessing differential expression measures and reproducibility in microarray assays. *Genome Biology*, *3*(11), research0062.
53. Yang, Y. H., & Speed, T. (2002). Design issues for cDNA microarray experiments. *Nature Reviews. Genetics*, *3*(8), 579–588.
54. Yates, F. (1936). Incomplete randomized blocks. *Annals of Eugenics*, *7*, 121–140.
55. Yates, F. (1936). A new method of arranging variety trials involving a large number of varieties. *Journal of Agricultural Science*, *26*, 424–455.
56. Zakharkin, S. O., Kim, K., Mehta, T., Chen, L., Barnes, S., Scheirer, K. E., Parrish, R. S., Allison, D. B., & Page, G. P. (2005). Sources of variation in Affymetrix microarray experiments. *BMC Bioinformatics*, *6*, 214.
57. Zhang, W., Carriquiry, A., Nettleton, D., & Dekkers, J. C. (2007). Pooling mRNA in microarray experiments and its effect on power. *Bioinformatics*, *23*, 1217–1224.

Chapter 9
The MicroArray Quality Control (MAQC) Project and Cross-Platform Analysis of Microarray Data*

Zhining Wen, Zhenqiang Su, Jie Liu, Baitang Ning, Lei Guo, Weida Tong, and Leming Shi

Abstract As a powerful tool for genome-wide gene expression analysis, DNA microarray technology is widely used in biomedical research. One important application of microarrays is to identify differentially expressed genes (DEGs) between two distinct biological conditions, e.g. disease versus normal or treatment versus control, so that the underlying molecular mechanism differentiating the two conditions maybe revealed. Mechanistic interpretation of microarray results requires the identification of reproducible and reliable lists of DEGs, because irreproducible lists of DEGs may lead to different biological conclusions. Many vendors are providing microarray platforms of different characteristics for gene expression analysis, and the widely publicized apparent lack of intra- and cross-platform concordance in DEGs from microarray analysis of the same sets of study samples has been of great concerns to the scientific community and regulatory agencies like the US Food and Drug Administration (FDA). In this chapter, we describe the study design of and the main findings from the FDA-led MicroArray Quality Control (MAQC) project that aims to objectively assess the performance of different microarray platforms and the advantages and limitations of various competing statistical methods in identifying DEGs from microarray data. Using large data sets generated on two human reference RNA samples established by the MAQC project, we show that the levels of concordance in inter-laboratory and cross-platform comparisons are generally high. Furthermore, the levels of concordance largely depend on the statistical criteria used for ranking and selecting DEGs, irrespective of the chosen platforms or test sites. Importantly, a straightforward method combining fold-change ranking with a non-stringent *P*-value cutoff produces more reproducible lists of DEGs than those by t-test *P*-value ranking. Similar conclusions are reached when microarray data sets

*The views presented in this article do not necessarily reflect those of the US Food and Drug Administration.

L. Shi (✉)
National Center for Toxicological Research, US Food and Drug Administration, 3900 NCTR Road, Jefferson, AR 72079, USA
e-mail: leming.shi@fda.hhs.gov

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_9,
© Springer-Verlag Berlin Heidelberg 2011

from a rat toxicogenomics study are analyzed. The availability of the MAQC reference RNA samples and the large reference data sets provides a unique resource for the gene expression community to reach consensus on the "best practices" for the generation, analysis, and applications of microarray data in drug development and personalized medicine.

9.1 Microarray Platforms for Genome-Wide Gene Expression Analysis

DNA microarray technology has been widely used in biomedical research as a high-throughput tool for simultaneously detecting the expression of thousands of genes [20, 32, 36]. By arraying tens of thousands of gene-specific DNA oligonucleotide probes on a solid surface such as a glass, plastic or silicon chip, a DNA microarray can hybridize the target sample from a well-defined condition to accomplish the equivalent number of genetic tests in parallel. Many microarray systems (platforms) are available and generally have quite different technical characteristics and fabrication procedures. In a recent review article [36], Shi and colleagues summarized the commonly used microarray platforms into three categories: (1) *in situ* synthesis of oligonucleotide probes on microarrays (e.g. Affymetrix GeneChip® microarrays with photolithography synthesis and Agilent's microarrays using inkjet synthesis); (2) spotting of pre-synthesized oligonucleotide probes on microarrays (e.g. GE Healthcare's CodeLink system, Applied Biosystems' Genome Survey Microarrays, and many forms of custom microarrays printed on glass slides with different sets of pre-synthesized oligonucleotides); and (3) deposition of pre-synthesized oligonucleotide probes on microsphere or bead based microarrays (Illumina's BeadChip microarrays). Different microarray platforms produce different types of errors and require different types of quality control and data analysis.

The Affymetrix GeneChip microarrays are arguably the most widely used platform in gene expression studies. By using photolithography, the DNA oligonucleotide probes are fabricated on a quartz wafer coated with a light-sensitive chemical. Each probe is usually 25 nucleotides long. The whole manufacturing process integrates semiconductor fabrication, solid-phase chemical synthesis, combinatorial chemistry, and molecular biology [6, 20]. Affymetrix GeneChip microarrays use 22 probes (or 11 pairs of probes), called a probeset, to measure the expression of a transcript. Half of the probes are perfect matches (PM) to a transcript and the other half are mismatches (MM) to the corresponding PM probes. A PM probe provides the fluorescence measurement for the target sample binding to it, whereas the paired MM probe provides the means for estimating non-specific fluorescence in the measurement. Many algorithms (e.g. MAS 5.0, dCHIP, RMA, PLIER, and other variants) are available to summarize probe-level data into probeset-level data that correlate with transcript abundance [16].

Agilent manufactures its microarrays using Hewlett-Packard's non-contact inkjet printing technology for *in situ* oligonucleotide synthesis [13]. Nucleotides are printed on a glass wafer, which is coated with a hydrophobic surface with exposed hydroxyl groups, by using the standard phosphoramidite synthesis. Each time only one nucleotide is printed at the location of the probe. This nucleotide printing process includes de-tritylation, oxidation and washing, and is repeated 60 times in order to create oligonucleotide probes 60 bases in length. The Agilent platform can be used in a two-color or one-color hybridization mode.

GE Healthcare's CodeLink microarray platform [29] utilizes a 3D-Link activated slides for arraying 30-mer probes. The Human Whole Genome Bioarray gives coverage of the human genome with over 50,000 transcripts and ESTs, including about 45,000 well-characterized human gene and transcript targets. The CodeLink products were discontinued in April, 2007.

Applied Biosystems uses standard phosphoramidite chemistry and solid-phase synthesis to pre-synthesize 60-mer oligonucleotides as probes for its microarrays [43]. The probes are deposited and covalently bound at the $3'$ end onto the microarray's derivatized nylon substrate that is, in turn, bound to a glass slide. A CCD camera is used for acquisition of chemiluminescent signals.

Illumina's BeadChip microarray is based on random self-assembly microspheres that are placed beforehand in millions of highly ordered microwells [10]. Different microarray types from Illumina contain microspheres targeting different numbers of transcripts, e.g. the HumanRef-8 Expression BeadChip measures more than 24,000 transcripts, whereas the Human-6 Expression BeadChip detects more than 48,000 transcripts. The HumanRef-8 and the Human-6 Expression BeadChips are considered as "arrays of arrays" because there are eight microarrays on one HuamnRef-8 chip and six microarrays on one Human-6 chip, thereby allowing researchers to simultaneously hybridize multiple samples on one chip at a more affordable cost per sample.

As a powerful tool for genome-wide gene expression analysis, DNA microarrays have been identified by the US FDA's Critical Path Initiative (http://www.fda.gov/oc/initiatives/criticalpath/) as a methodology to advance drug development and personalized medicine through the identification of biomarkers. Indeed, many microarray based pharmacogenomics data sets have been submitted by the industry to the US FDA [8]. One important application of microarrays is to identify differentially expressed genes (DEGs) between two distinct biological conditions, e.g. disease versus normal, treatment versus control, or safety versus adverse reactions, so that the underlying molecular mechanism differentiating the two conditions maybe revealed [36].

9.2 Statistical Methods for Identifying Differentially Expressed Genes (DEGs)

Although DEGs are important to the genomics analysis with microarrays, it is still a pendent problem regarding how to best determine what genes are significantly differentially expressed between two groups of samples. The reliability of gene expression measurement can be influenced by many factors such as techniques, instruments, and statistical methods. Any errors from these aspects can render microarray data unreliable or invalid. Therefore, it is not surprising that many statistical methods have been proposed to identify DEGs. The essence of identifying DEGs from microarray data is to rank all genes (transcripts) measured by the microarray according to a defined criterion and to set a more-or-less arbitrary cutoff of that criterion for gene selection [39]. Here, we briefly describe three frequently used gene selection methods in microarray data analysis.

9.2.1 Welch's T-Test P Value

The *P* value from Welch's t-test is one of the most widely used statistical methods for identifying DEGs. The null hypothesis of the test is that the two sample groups (e.g. control group and treatment group) come from the same sample population, and the t distribution is used to test this hypothesis. The statistic t is defined by Eq. 9.1:

$$t = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}} \tag{9.1}$$

where $\overline{x}_1$, $\overline{x}_2$ are the sample mean of groups 1 and 2, respectively; s_1, s_2 are the sample variance of groups 1 and 2, respectively; and N_1, N_2 are the number of samples in groups 1 and 2, respectively.

The Welch's t-test yields a *P* value that can be defined by Eq. 9.2:

$$p = 1 - \int_{-t}^{t} \frac{\Gamma(\frac{F+1}{2})}{\Gamma(\frac{F}{2})} \frac{1}{\sqrt{F \times \pi}} \frac{1}{(1 + \frac{t^2}{F})^{\frac{F+1}{2}}} dt \tag{9.2}$$

where F is the degree of freedom and can be estimated by Eqs. 9.3 and 9.4 with an assumption of equal or unequal variance, respectively:

$$F = N_1 - N_2 - 2 \tag{9.3}$$

$$F = \frac{(s_1^2/N_1 + s_2^2/N_2)^2}{(s_1^2/N_1)^2/(N_1 - 1) + (s_2^2/N_2)^2/(N_2 - 1)} \tag{9.4}$$

Given a *P* value threshold, such as 0.05, we can assume that the null hypothesis is true when the *P* value is greater than the threshold and the null hypothesis is false when the *P* value is less than the threshold. For each gene in a microarray study, a *P* value can be calculated by testing its expression in two sample groups. Genes with a *P* value less than the threshold are considered differentially expressed.

9.2.2 Fold Change (FC)

The fold change (or ratio) measures the magnitude of differential gene expression and is commonly used by biologists for analyzing gene expression data because of its simplicity and biological relevance [32]. In practice, there are two ways to calculate fold change. The first one is described by Eq. 9.5:

$$\log_2 FC_i = \frac{1}{N_1}\sum_{j=1}^{N_1}\log_2 x_{ij,1} - \frac{1}{N_2}\sum_{j=1}^{N_2}\log_2 x_{ij,2} \tag{9.5}$$

where $x_{ij,1}$ and $x_{ij,2}$ are the raw expression intensity values of the ith gene for the jth replicate in groups 1 and 2, respectively. N_1 and N_2 are the number of samples in groups 1 and 2, respectively.

The second definition of fold change is given by Eq. 9.6:

$$\log_2 FC_i = \log_2\left(\frac{\frac{1}{N_1}\sum_{j=1}^{N_1} x_{ij,1}}{\frac{1}{N_2}\sum_{j=1}^{N_2} x_{ij,2}}\right) \tag{9.6}$$

where $x_{ij,g}$ and Ng $(g = 1, 2)$ are the raw expression intensity values of the ith gene for the jth replicate and the number of samples in groups 1 and 2, respectively. For the same data set, the log2FC values calculated by Eqs. 9.5 and 9.6 for the same gene are usually very close. Equation 9.5 is commonly used when the gene expression data are already presented in a log2 transformed form so that the difference of the group means is the log2FC value.

After calculating the log2FC value for each gene, we can rank all the genes by the absolute of their log2FC values (|log2FC|) with descending order and select those genes whose |log2FC|values are greater than a given fold change cutoff, such as a FC of 2.0 or |log2FC|of 1.

9.2.3 Significance Analysis of Microarrays (SAM)

Significance Analysis of Microarrays (SAM), proposed by Tusher and colleagues in 2001 [41], is a statistical method by combining the gene specific t-test with a statistic d_i for each gene to determine whether the gene expression of a gene is significantly different between two groups of samples. The calculation procedure is based on a permutation analysis of gene expression data and can be presented in the following steps:

Suppose the raw expression data is x_{ij}, where $i = 1, 2, \ldots, p$ genes and $j = 1, 2, \ldots, n$ samples. The respond data is y_j, $j = 1, 2, \ldots, n$.

1. Calculate a statistic d_i by Eq. (9.7):

$$d_i = \frac{r_i}{s_i + s_0} \tag{9.7}$$

For the *i*thgene, r_i is the difference of the group means (i.e. log_2FC) for two-class comparisons; s_i is the standard deviation; and s_0 is an exchangeability factor.

1. Order the d_i values by their magnitudes.
2. Take k subsets of permutations of the response values y. For each permutation calculate the ordered d values.
3. Set a threshold and determine a gene as significantly differentially expressed when the absolute value of the test statistic d for that gene minus the mean test statistic $\overline{d}$ for that gene is greater than the threshold.
4. Estimate the false discovery rate based on the expected versus the observed values.

The R-package of SAM can be downloaded from the website: http://www-stat.stanford.edu/~tibs/SAM/.

9.3 The Challenge: Apparent Lack of Reproducibility of Differentially Expressed Genes (DEGs)

The critical issue regarding the reproducibility of DNA microarray results in terms of DEGs has been raised in the literature [40] and greatly publicized [21]. That is, with the same sets of testing RNA samples, the lists of DEGs identified from different platforms or laboratories or from different gene selection methods showed little overlap. For instance, only four common DEGs were found when analyzing identical sets of RNA samples with three commercial microarray platforms, Affymetrix, Agilent, and Amersham (i.e. GE Healthcare). Between the two lists of about 200 stem cell-specific genes separately identified by Ramalho-Santos et al. [30] and Ivanova et al. [17] with the Affymetrix platform under similar experimental conditions, only six DEGs were found in common. From the 138 and 425 DEGs identified from Affymetrix and CodeLink platforms, respectively, in a neurotoxicological study by Miller et al. [28], the number of overlapped DEGs is only 11. All these studies use t-test P values to rank genes and create DEG lists by setting a P threshold. The percentage of overlapping genes (POG) is used as the measure of inter-site or cross-platform reproducibility of DEGs.

With criticisms and concerns about the reproducibility of microarrays appeared in the peer-reviewed journals [7, 14, 21, 26, 27, 40], the growing negative perception about the reliability of microarray platforms had been widespread and caused much discussion. Some explanations and solutions have been proposed to address the apparent lack of reproducibility of DEG lists from microarray data. Ein-Dor et al. suggested increasing the sample size [5]; Allison et al. discussed the pros and cons of different statistical methods for selecting differentially expressed genes [1]; Mecham et al. increased the consistency and reproducibility in microarray results by mapping probe sequences across platforms [22]; Hoffman et al. improved the quality of expression profiling by fully standardizing sample preparation and hybridization procedures [12].

Shi et al. [33] reanalyzed the data published by Tan et al. [40] and concluded that the lack of reproducibility of the DEG lists greatly depends on the choice of statistical methods for selecting DEGs. Using the same data set with different data analysis and gene selection criteria, the cross-platform concordance was largely improved. Specifically, when the DEG lists were determined by simply using either SAM or fold change (FC) with noise filtering, the cross-platform concordance was increased by five to nine folds. These observations raised the awareness that microarray cross-platform reproducibility in terms of DEGs is sensitive to the choice of gene selection methods and in fact became a major motivator for the launch of the MAQC project [33].

9.4 The MicroArray Quality Control (MAQC) Project

Many vendors are providing microarray platforms of different characteristics for gene expression analysis, and the widely publicized apparent lack of intra- and cross-platform concordance in DEGs from microarray analysis of the same sets of study samples has been of great concerns to the scientific community and the US Food and Drug Administration (FDA). The FDA-led MicroArray Quality Control (MAQC) project aims to objectively assess the performance of different microarray platforms and the advantages and limitations of various competing statistical methods in identifying DEGs from microarray data. As illustrated in Fig. 9.1, in this chapter we describe the study design of and the main findings from the MAQC project by using large data sets generated on human reference RNA samples established by the MAQC project. We show that the levels of concordance in inter-laboratory and cross-platform comparisons are generally high. Furthermore, the levels of concordance largely depend on the statistical criteria used for ranking and selecting DEGs, irrespective of the chosen platforms or test sites. Importantly, a straightforward method combining fold-change ranking with a non-stringent *P*-value cutoff produces more reproducible lists of DEGs than those by t-test *P*-value ranking. These conclusions are verified when microarray data sets from a rat toxicogenomics study are analyzed.

It should be noted that microarray results can be presented in several ways including absolute intensity (signal), relative intensity (ratio or fold change), DEGs, Gene Ontology, or pathways. In this chapter, we focus on inter-site and cross-platform data comparisons in terms of DEGs because they usually form the foundation of biological interpretation of microarray results by biologists and the apparent lack of reproducibility in DEGs has been used as evidence to question the reliability of microarray technology.

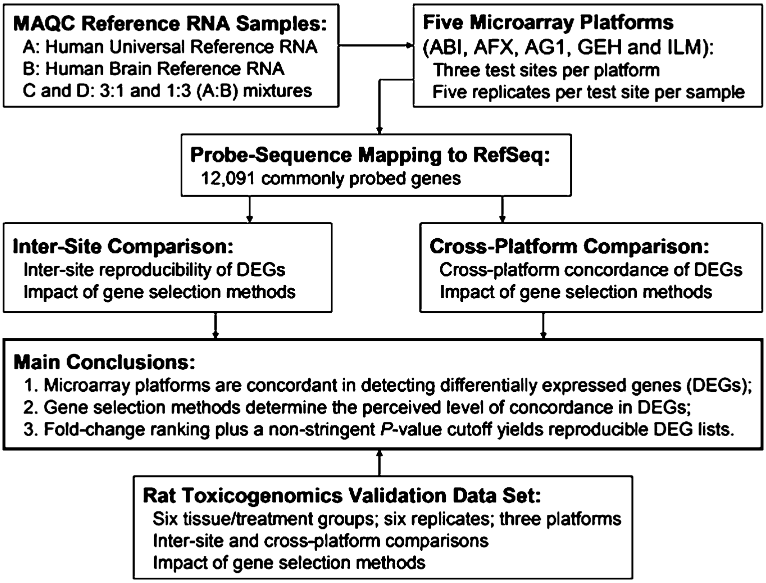

MAQC Reference RNA Samples:
A: Human Universal Reference RNA
B: Human Brain Reference RNA
C and D: 3:1 and 1:3 (A:B) mixtures
Five Microarray Platforms
(ABI, AFX, AG1, GEH and ILM):
Three test sites per platform
Five replicates per test site per sample
Probe-Sequence Mapping to RefSeq:
12,091 commonly probed genes
Inter-Site Comparison:
Inter-site reproducibility of DEGs
Impact of gene selection methods
Cross-Platform Comparison:
Cross-platform concordance of DEGs
Impact of gene selection methods
Main Conclusions:
1. Microarray platforms are concordant in detecting differentially expressed genes (DEGs);
2. Gene selection methods determine the perceived level of concordance in DEGs;
3. Fold-change ranking plus a non-stringent P-value cutoff yields reproducible DEG lists.
Rat Toxicogenomics Validation Data Set:
Six tissue/treatment groups; six replicates; three platforms
Inter-site and cross-platform comparisons
Impact of gene selection methods

Fig. 9.1 The experimental design of the MAQC project and main microarray data sets and analyses used in this chapter

9.4.1 The MAQC Study Design: Two Reference RNA Samples, Five Microarray Platforms, Three Test Sites per Platform, and Five Replicates per Test Site

For the purpose of objectively assessing the performance of DNA microarrays and the advantages and limitations of various data analysis methods for identifying DEGs, the MAQC project was launched in 2005 and is a truly community-wide effort involving 137 participants from 51 organizations [34]. In the project, as shown in Fig. 9.1, two high-quality, distinct human reference RNA samples, along with two mixtures of defined ratios, were prepared for assessing the repeatability of gene expression microarray data within a specific site, the reproducibility across multiple sites, and the concordance across multiple platforms [34]. Each microarray platform was tested at three independent test sites and each RNA sample was replicated five times at each test site. The two RNA samples are a Universal Human Reference RNA (UHRR) from Stratagene and a Human Brain Reference RNA (HBRR) from Ambion. Although these two samples do not directly represent a relevant biological study such as control versus treatment, the MAQC data sets with the two reference RNA samples indeed provide important technical insights into the capabilities and limitations of the microarray technology and the corresponding data analysis approaches.

The major microarray platforms tested in the MAQC project included Applied Biosystems (ABI), Affymetrix (AFX), Agilent Technologies (AG1), GE Healthcare

(GEH), and Illumina (ILM), and cover a wide range of technical characteristics, as summarized in Table 9.1. Compared to what are presented in this chapter, the original MAQC project included more platforms, more test sites, and more sample types. The entire MAQC data sets are available from NCBI GEO (GSE5350) or at the MAQC Web site (http://edkb.fda.gov/MAQC/MainStudy/upload/).

9.4.2 Probe-Sequence Based Mapping to the RefSeq Reference Transcriptome

Different microarray platforms are based on distinct probe-design strategies and manufacturing processes and are targeting different subsets of the whole human transcriptome. To allow for cross-platform comparison of gene expression results, the first challenging task is to identify the subset of genes (transcripts) that are commonly measured by all microarray platforms under comparison. The annotation information provided by the microarray vendors is usually based on different mapping strategies from probes to genes (transcripts). Thus, the MAQC project requested the probe sequences from each platform provider and mapped the probe sequences to the same transcriptome reference database, i.e. the March 8, 2006 version of human RefSeq release containing about 24,000 curated accessions (http://www.ncbi.nlm.nih.gov/RefSeq). For a probe to be considered as a match to a transcript, its sequence was required to perfectly match the sequence of the database entry. Probes matching only the reverse strand of a transcript were excluded as well as probes matching more than one gene. For the Affymetrix platform, an exact match of 80% of the probes within a probeset (usually 9 probes out of 11) was required. As shown in Table 9.1, each of the high-density microarray platforms measures a similar number of RefSeq transcripts (20,230–22,161) and a similar number of Entrez genes (15,429–16,990). Finally, 15,615 RefSeq entries are measured on all of the high-density microarray platforms, representing 12,091 unique Entrez genes. To simplify the inter-site and cross-platform comparisons, we created a “one-probe-to-one-gene” list with 12,091 probes from each platform and the corresponding 12,091 reference sequences from 12,091 different genes. Results shown below on the MAQC reference data sets are based on these 12,091 “common” genes.

9.4.3 Percentage of Overlapping Genes (POG) for Assessing Reproducibility of DEGs

The Percentage of Overlapping Genes (POG) [33, 34, 37] was used as a metric for assessing the reproducibility in two types of comparisons: (1) inter-site comparison using data from the three test sites with the same platform; and (2) cross-platform comparison between ABI, AFX, AG1, GEH, and ILM using data from their fist test sites.

Table 9.1 Summary of major microarray gene expression platforms used in the MAQC project and probe-sequence based mapping to RefSeq

Manufacturer	Platform	Code	Detection method	Type	Probe length	Number of probe(set)s	Number of analyzed probe sequences	Number of mapped probes	Number of RefSeq NM accessions mapped to probes	Number of Entrez gene ID's mapped to probes
Applied Biosystems	Human Genome Survey Microarray v2.0	ABI	Chemiluminescence	Presynthesized Oligos	60	32,878	32,878	18,547	21,963	16,763
Affymetrix	HG-H133 Plus 2.0 GeneChip®	AFX	Fluorescence	*In situ* synthesis	25	54,675	54,675	24,694	21,318	15,965
Agilent Technologies	Whole Human Genome Oligo Microarray, G4112A	AG1	Fluorescence	*In situ* synthesis	60	43,931	41,000	22,677	21,890	16,493
Ge Healthcare	CodeLink™ Human Whole Genome	GEH	Fluorescence	Presynthesized Oligos	30	54,359	53,423	16,881	20,230	15,429
Illumina	Human-6 BeadChip, 48 k v1.0	ILM	Fluorescence	Presynthesized Oligos on BeadChip	50	47,293	47,282	20,140	22,161	16,990
National Cancer Institute	Operon Human Oligo Set v3	NCI	Fluorescence	Presynthesized Oligos	39–70	37,635	35,235	21,555	20,987	15,899
						Union	264,493	125,216	23,971	18,114
						Intersection			15,615	**12,091**

Notes: The number of probes for which mapping was attempted may slightly differ from the number of probes arrayed (Table 9.1) because of the removal of control probes and replicate spots. An exact sequence match was required and probes that match more than one gene were excluded. For the AFX platform, there are generally 11 perfect-match probes per probeset, and each probe was mapped individually. An exact match of 80% of the probes in a probeset was required for the probe set to qualify as a perfect match. The common set of 12,091 genes is represented on the six high-density microarray platforms and used for cross-platform comparison. The two-color microarray data from the NCI platform are not used in this chapter. Additional microarray and alternative gene expression platforms were used in the MAQC project [34]; however, for simplicity of description they are not discussed in this chapter.

For two lists of DEGs, POG is calculated as follows: POG $= 100 * (\mathrm{DD} + \mathrm{UU})/2\mathrm{L}$, where DD and UU are the number of commonly down- or up-regulated genes, respectively, from the two DEG lists, and L is the number of genes selected from the up- or down-regulation directionality. To overcome the confusion of different numbers for the denominator, in our POG calculations we deliberately selected an equal number of up-regulated and down-regulated genes, L. The POG graphs shown in this chapter are essentially the same as the CAT (correspondence at the top) plots introduced by Irizarry et al. [15] except that in the current POG graphs the x-axis is in log-scale so that the details when fewer genes are selected can be more easily seen.

The number of DEGs from a given study can vary as the threshold for gene selection can be chosen arbitrarily. Therefore, we calculated POG values for many different cutoffs. The number of genes available for ranking and selection in one direction, L, varies from 1 to 6,000 (with a step of one) or when there are no more genes in one regulation direction, corresponding to 2L varying from 2 to 12,000. If a gene is selected by two test sites or platforms but is in different regulation directionalities, it is considered as discordant. Therefore, in reality POG can hardly reach 100%.

We considered six gene ranking (selection) methods: (1) FC (fold change ranking); (2) FC_P0.05 (FC-ranking with *P* cutoff of 0.05); (3) FC_P0.01 (FC-ranking with *P* cutoff of 0.01); (4) *P* (*P*-ranking, simple t-test assuming equal variance); (5) P_FC2 (*P*-ranking with FC cutoff of 2); (6) P_FC1.4 (*P*-ranking with FC cutoff of 1.4). The platform manufacturers' recommended normalization methods are used for data pre-processing: PLIER16 for Affymetrix, median scaling for ABI, Agilent, and GE Healthcare, and quantile normalization for ILM.

9.4.4 Inter-Site Concordance with Data from MAQC Reference RNA Samples

Figure 9.2 plots the inter-site POG versus the number of genes selected as differentially expressed. Since there are three possible inter-site comparisons for the same platform (S1–S2, S1–S3, and S2–S3, where S = Site) and six gene selection methods, there are 18 POG lines for each platform in the inter-site comparison. It is clear that for some gene selection methods the inter-site reproducibility heavily depends on the number of genes chosen as differentially expressed (when x-axis moves from the left to the right). In addition, the gene ranking criterion also greatly impacts the perceived POG: Gene selection using FC-ranking leads to higher POG than *P*-ranking. The inter-site POG from FC-ranking is near 90% for as few as 20 genes for most platforms, and remains almost the same level as the number of selected genes increases. In contrast, the inter-site POG from *P*-ranking is in the range of 20–40% for as many as 100 genes, and then approaches 90% only after several thousand genes are selected. Because microarray technology is widely used for identify biomarkers consisting of a relatively small number of genes, *P*-ranking could

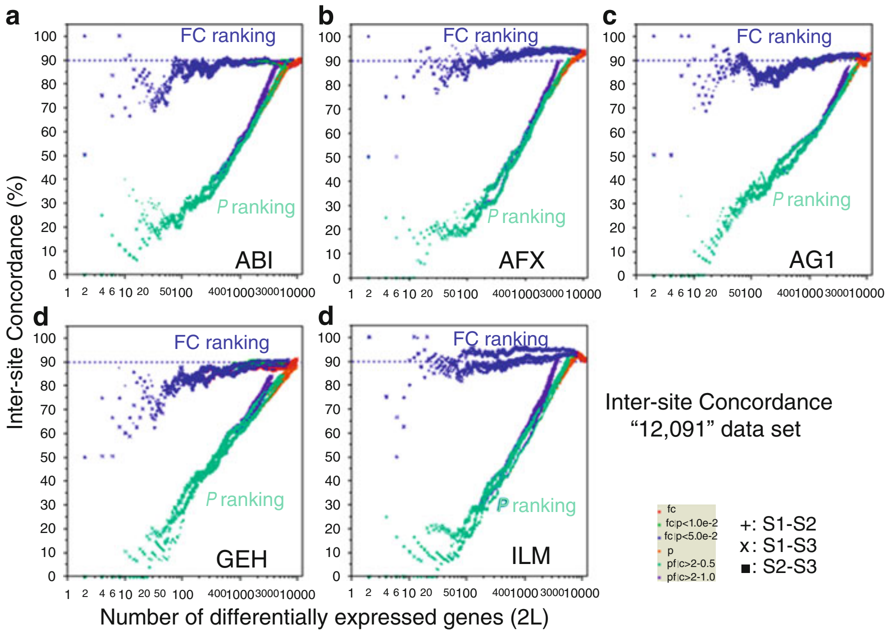


Fig. 9.2 Concordance of inter-site comparisons with data generated on MAQC reference RNA samples A and B. Each panel represents results of inter-site consistency for a commercial platform in terms of overlap of DEGs. For each platform and each gene selection method, there are three possible inter-site comparisons: S1-S2, S1-S3, and S2-S3. Therefore, each panel consists of 18 POG lines that are colored based on gene ranking/selection method. The x-axis represents the number of selected DEGs, and the y-axis is the percentage (%) of genes common to the two gene lists derived from two test sites at a given number of DEGs. (Reproduced with permission from Shi L et al. [37])

leave the impression that microarray data are not reproducible between different test laboratories when smaller numbers of genes are selected as differentially expressed.

9.4.5 Cross-Platform Concordance with Data from MAQC Reference RNA Samples

Figure 9.3 shows the dependence of cross-platform concordance on the number of genes selected as differentially expressed and on the gene selection methods. For each gene selection method, there are ten ($10 = 5 \times 4/2$) cross-platform pairs for comparison between the five platforms. Similar to inter-site comparisons shown in Fig. 9.2, *P*-ranking leads to lower cross-platform POG than FC-ranking. When FC is used to rank and select DEGs from each platform, the cross-platform POG is around 70–85%, depending on the platform pair and the number of DEGs selected. The decrease in concordance from inter-site comparison (Fig. 9.2) to

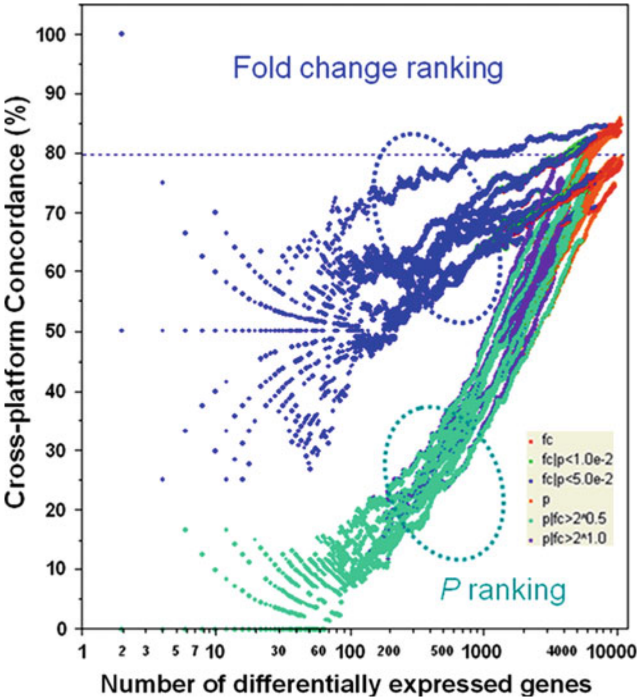


Fig. 9.3 Cross-platform concordance of DEG lists based on data from MAQC reference RNA samples A and B. For each platform, the data from test site 1 are used for cross-platform comparison. Each POG line corresponds to comparison of the DEGs from two microarray platforms using one of the six gene selection methods. The x-axis represents the number of selected DEGs, and the y-axis is the percentage (%) of genes common to the two gene lists derived from two platforms at a given number of DEGs. POG lines circled by the *blue oval* are from fold change based gene selection methods, whereas POG lines circled by the *teal oval* are from P based gene selection methods. (Reproduced with permission from Shi L et al. [37])

cross-platform comparison (Fig. 9.3) reflects the inherent technological differences between different microarray platforms in addition to inter-site differences.

For the analysis of a typical microarray data set, the number of DEGs is determined by setting a more-or-less arbitrary cutoff of the ranking criterion. To illustrate the typical level of concordance of DEGs from inter-site and cross-platform comparisons, we considered a typical scenario where a gene is considered as differentially expressed when the t-test *P* value is <0.001 and the fold change is 2. The DEG list from each test site is compared with the lists from the other two test sites of the same platform and those from test sites using other microarray platforms. The percentage of overlap in DEGs in each comparison is displayed in Fig. 9.4. The gene list overlap is more than 60% for each pair of platforms. Many test site pairs achieve 80% or more overlap within and between platforms.

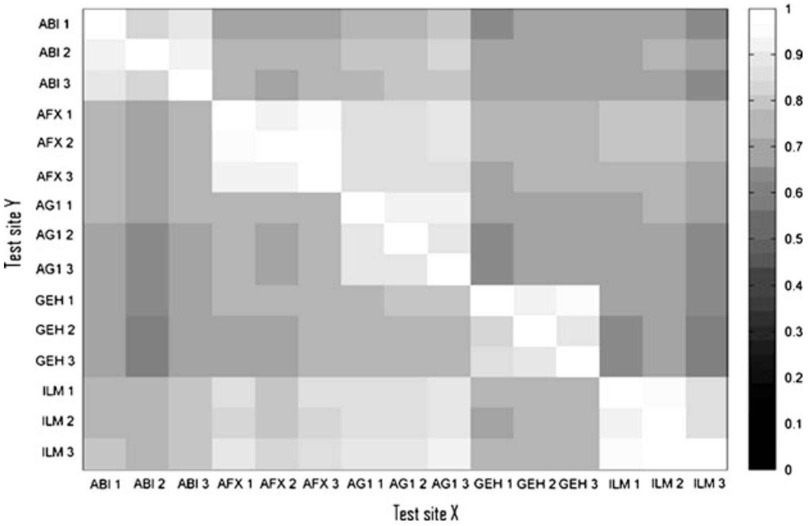


Fig. 9.4 Percentage of overlapped DEGs between two different test sites or platforms. The DEGs are selected from each test site from the 12,091 commonly measured genes with fold change value >2 and P value <0.001. The agreement between any two test sites using the same platform is about 90%, whereas the concordance between any two test sites using different platforms is at least 60%. Note that this graph is not symmetrical along the diagonal line because the percentage of test site Y genes on the list from test site X (*the upper right triangle*) can be different from the percentage of test site X genes on the test site Y list (*the lower left triangle*)

Among the 12,091 "common" genes measured by all genome-wide microarray platforms within the MAQC project, 906 genes were also analyzed by the "gold standard" TaqMan PCR gene expression assays [2, 34]. The concordance in DEGs between microarrays and TaqMan was around 80%, indicating a high level of reliability of microarray data [37]. Consistent with inter-site and cross-microarray platform comparisons, the POGs comparing microarrays with TaqMan also depend on the choice of the ranking criteria for gene selection, and FC ranking results in markedly higher POG than ranking by *P* alone, especially for short gene lists [37].

9.4.6 Gene Selection Methods Determine the Level of Reproducibility of Microarray Results

Figure 9.5a shows how inter-site reproducibility is impacted by the choice of different statistical methods including FC-ranking, t-test statistic, and SAM when results using MAQC reference RNA samples from Affymetrix test sites 1 and 2 are compared. The POG for SAM is greatly improved over that of t-test statistic, but does not exceed the level of POG based on FC-ranking. Similar findings (Fig. 9.5b) are observed using the rat toxicogenomics data set of Guo et al. [11].

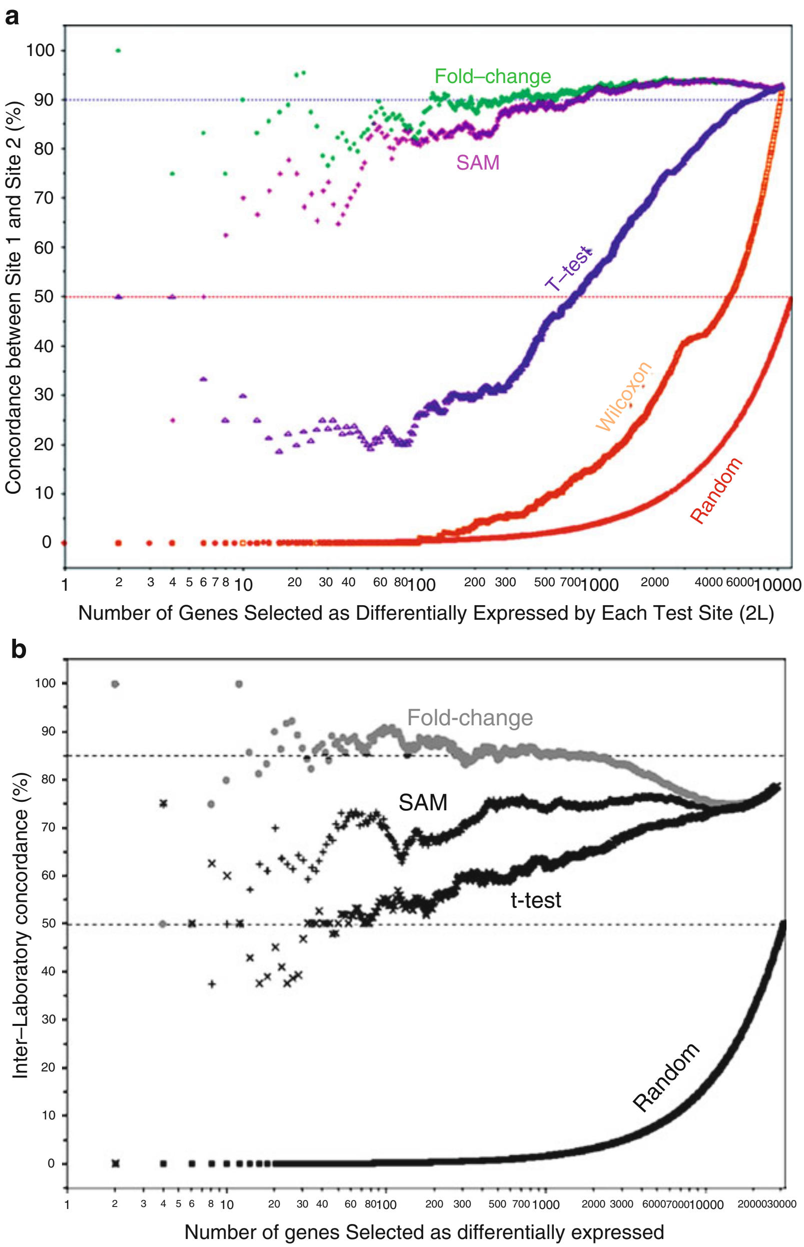


Fig. 9.5 The level of inter-site concordance depends on the choice of gene selection methods. *Panel a*: Affymetrix data generated on MAQC reference RNA samples A and B; *Panel b*: Affymetrix data generated from a rat toxicogenomics study (comfrey treatment versus controls). The x-axis represents the number of selected DEGs, and the y-axis is the percentage (%) of genes common to the two gene lists derived from two test sites at a given number of DEGs. (Reproduced with permission from Shi L et al. [37] and Guo L et al. [11])

9.4.7 The Rat Toxicogenomic Study: A Validation of Reproducibility of Microarray Results

The high concordance of the five microarray platforms in the MAQC project described above and the impact of gene selection methods on the reproducibility of the resulting DEG lists are obtained using the two human reference RNA samples that lack explicit biological connections. Questions remain whether microarray data from different laboratories or platforms will achieve similar results when real-world biological RNA samples are used. In order to validate the consistency across different microarray platforms and test laboratories and to further investigate the impact of different statistical approaches on the reproducibility of DEG lists, we generated a rat toxicogenomics data set. Briefly, 36 RNA samples are isolated from the kidney and/or liver of rats exposed to one of the three botanical carcinogens, namely aristolochic acid (AA), riddelliine (RDL), and comfrey (CFY) [4, 23–25], representing six treatment/tissue groups: kidney from aristolochic acid-treated rats, kidney from vehicle control rats, liver from aristolochic acid-treated rats, liver from riddelliine-treated rats, liver from comfrey-treated rats, and liver from vehicle control rats. There are six biological replicates within each treatment/tissue group. For the purpose of cross-platform comparison, these 36 samples are hybridized to four microarray platforms: Affymetrix, Agilent, Applied Biosystems, and GE Healthcare. In addition, the Affymetrix platform is tested at two different sites to evaluate the level of inter-site reproducibility. PLIER16 data are used for Affymetrix and median scaled data are used for ABI, Agilent, and GE Healthcare.

The reproducibility between the two test sites using the Affymetrix platform is shown in Fig. 9.6. For each treatment versus tissue-type matched control comparison, the inter-site concordance in DEGs is as high as 80–90% when DEGs are selected by FC ranking and the number of genes selected as differentially expressed ranges from a few to ∼2,000. With more genes considered as differentially expressed, the inter-site concordance began to decline because more genes with smaller fold change values are included in the gene lists. On the contrary, the concordance of the gene lists is low when the t-test *P* value is used for ranking and selecting DEGs, in particular when a smaller number of genes are selected.

Cross-platform concordance is assessed using comfrey (CFY) treatment versus control as an example. The probe sequences of the four rat microarrays were mapped to the rat RefSeq database (March 2006) using the same criteria described above for the human microarrays, resulting in 5,112 genes commonly measured by the four rat microarray platforms. The trend of the results is similar: fold change ranking generates more reproducible DEG lists across platforms (Fig. 9.7). The cross-platform concordance is 50–70% and increases to 70–80% when “flagged” (i.e. undetectable) genes are excluded from the analysis.

One important task in microarray data analysis is to provide mechanistic interpretation to the DEGs. Using the rat toxicogenomics data set, the MAQC project demonstrated that more reproducible DEG lists lead to more consistent biological interpretation of microarray results in terms of gene ontology terms or pathways enriched by the DEGs [11]. These results indicate that the biological interpretation

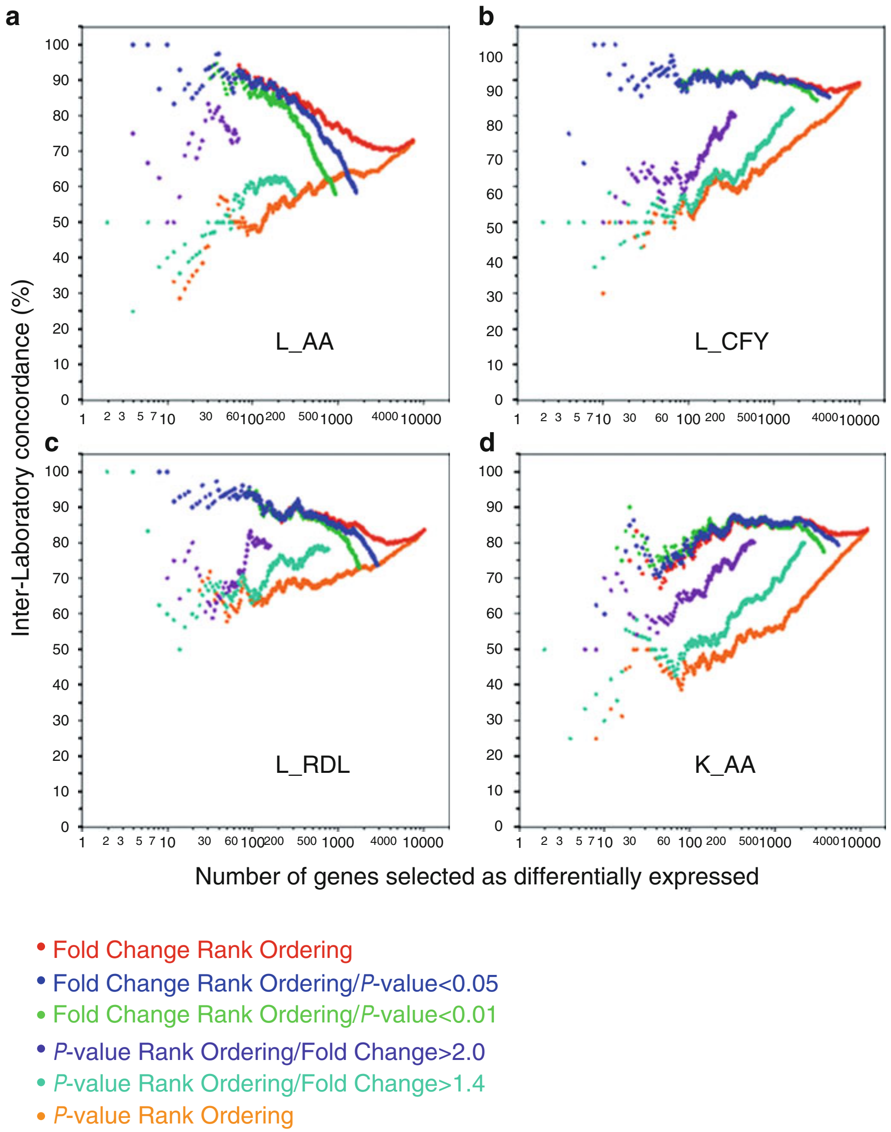


Fig. 9.6 Inter-laboratory concordance of DEG lists selected from four treatment-to-control comparisons. *Panel a*: liver from aristolochic acid (AA) treatment versus liver control; *Panel b*: liver from comfrey (CFY) treatment versus liver control; *Panel c*: liver from riddelliine (RDL) treatment versus liver control, and *Panel d*: kidney from aristolochic acid (AA) treatment versus kidney control. The x-axis indicates the number of genes selected as differentially expressed and the y-axis represents the overlap percentage of the DEG lists from two test sites. (Reproduced with permission from Guo L et al. [11])

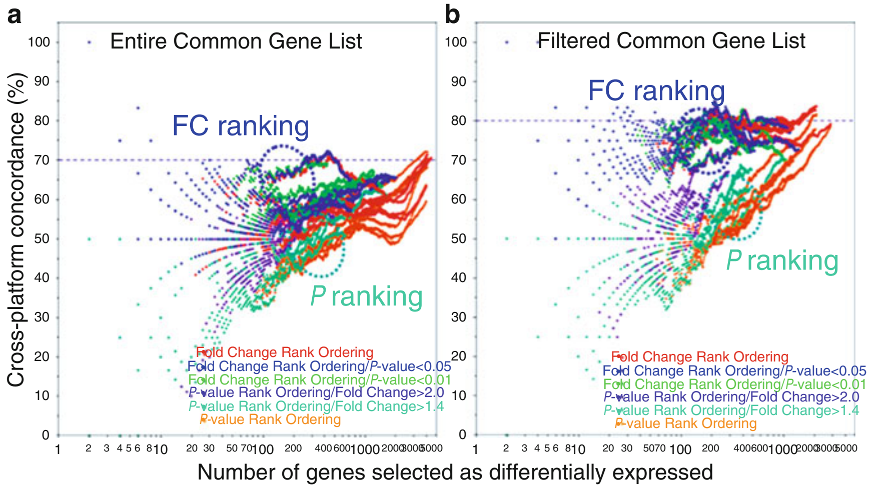


Fig. 9.7 Cross-platform concordance of DEG lists based on data from a rat toxicogenomics study. For each comfrey treatment to control comparison, the percentage of overlap of differentially expressed genes was calculated using different selection methods from the 5,112 genes commonly measured by the four platforms (ABI, AFX, AG1, and GEH). The x-axis represents the number of genes selected as differentially expressed, and the y-axis details the overlap (%) of two gene lists for a given number of differentially expressed genes. The results depicted are derived from the comfrey-treated comparisons for each platform, but similar results were generated with the other treatment comparisons. Differentially expressed genes are selected from either the entire common gene list (*panel a*) or after filtering of low intensity or non-detectable genes and identifying the subset of common genes that were detectable by both comparisons (*panel b*). (Reproduced with permission from Guo L et al. [11])

of microarray data will be more reproducible when DEGs are selected with fold change based ranking that produces more stable DEG lists.

9.5 Conclusions and Discussion

Microarray technology has had a profound impact on biological research partially due to its ability to identify differentially expressed genes that may be used to develop potential biomarkers, elucidate molecular mechanisms, and group similar samples based on gene signatures. Therefore, the reproducibility and reliability of the DEG lists created by gene selection criteria are critical for biological interpretation. To address the issue of microarray reproducibility, we have designed an experiment with well-controlled conditions and performed inter-site and cross-platform comparisons in the MAQC project using different statistical methods for gene selection.

Most of the previous studies questioning the reproducibility and reliability of microarrays for gene expression analysis are based on the statistical significance

(*P* value) alone instead of the actual measured quantity of differential expression (fold change or ratio) for selecting DEGs. The reliance on only *P* value to create DEGs lists has resulted in the apparent irreproducibility between test sites and between microarray platforms. Our results from analyzing data sets from the MAQC human reference RNA samples and the rat toxicogenomics study samples indicate that a straightforward approach of fold change ranking combined with a non-stringent *P* value cutoff can successfully generate reliable DEG lists. Furthermore, compared to *P* value ranking, this joint method can minimize the impact of normalization methods on the reproducibility of DEGs lists. That is, the DEG list from *P* value ranking based gene selection methods is more susceptible to the choice of normalization methods [11]. We recommend a straightforward approach of fold change ranking combined with a non-stringent *P* value cutoff as a baseline practice for microarray data analysis to generate reproducible lists of DEGs. The fold change criterion ensures the reproducibility of DEGs and the *P* value criterion controls false positives.

Many statisticians including those originally involved in analyzing the MAQC data sets were puzzled by the finding that fold-change ranking combined with a non-stringent *P*-value cutoff produced more reliable lists of DEGs than *P* value alone based methods. However, if we consider microarrays as one bioanalytical measurement technique, then the MAQC finding is in fact not unexpected. For differential gene expression measurements, what is actually being measured by microarrays is the difference in gene expression levels for the same gene between two sample groups. For a biologist, the difference is of course the fold change (or ratio) of the expression levels between the two groups. Whether the detected fold change is statistically significant or relies on the *P* value of a statistical test such as t-test. For genes meeting a statistical significance threshold (e.g. $P < 0.05$), a larger fold change is easier to be more reliably detected by microarrays because the signal (fold change) is stronger. On the other hand, the *P* value reflects the signal-to-noise ratio. Even though the signal (fold-change) may be reproducible between laboratories or platforms, the noise in microarray measurements are difficult to characterize and can vary dramatically between laboratories or platforms. That is, noise cannot be reproduced. Therefore, it is not surprising that the *P* values are not as reproducible as the fold changes between laboratories or platforms. It should be pointed out that genes with the largest fold changes are not necessarily the most biologically important ones in a specific study. In fact, those with small fold changes may play a critical role; but, unfortunately, such genes are hard to be identified by microarray or other gene expression technologies, because the signal (fold change) is too low compared to the detection limit of the measurement technology.

Statistical metrics such as sensitivity and specificity are routinely used in assessing the performance of gene selection methods where a pre-defined "truth" about differentially gene expression is required. However, it could be argued that every gene in a microarray study is differentially expressed depending on the chosen threshold, because a gene can rarely have exactly the same expression level in two sample groups. Thus, one critical step in a microarray study is to identify a subset of genes that are more reliably detectable as differentially expressed. Reproducibility is a critical dimension to consider along with sensitivity and specificity when defining a gene list, but it is seldom emphasized. Because irreproducibility has rendered microarray technology vulnerable to criticism, the analyses in the MAQC project emphasized on reproducibility of the gene lists.

The levels of inter-site and cross-platform concordance reached in the MAQC project are not necessarily achievable in other studies. We note that data quality is critically important to ensure data reproducibility. In addition, the degree of differences between the two groups of samples being compared plays an important role in determining the level of concordance. Genes with small magnitude of differential gene expression (i.e. fold change) in a microarray platform are less likely to be consistently identified as differentially expressed in another laboratory or by another platform, and under such scenarios of noisy data, no gene selection methods can guarantee a high level of reproducibility of DEGs [37]. Many biologically important genes in brain tissues may only exhibit small levels of differential expression. Therefore, we routinely use more than one microarray platform to increase the reliability of the identified DEGs using non-stringent P and fold change cutoffs to select DEGs from each platform. Understandably, in such real-life studies many genes with small fold changes close to the detection limit of microarray technologies are considered as DEGs on a single platform. However, we reason that genes with small fold changes but of truly biological significance should be more likely to be detected as DEGs by two independent microarray technologies (e.g. Affymetrix and Agilent), thereby decreasing false positives in microarray analysis. With the cost for microarray analysis per sample continues to fall, using multiple microarray platforms to reliably detect differentially expressed genes is becoming more and more feasible.

The two human reference RNA samples used in the MAQC project are commercially available and have been widely used by the gene expression community to evaluate the performance of new technologies such as nanostring [9] or next-generation sequencing [42] and to ensure laboratory proficiency. The availability of the large reference microarray and TaqMan data sets from the MAQC project provide a unique resource to the scientific community to objectively assess the advantages and limitations of different data analysis approaches [18].

The MAQC project, while positively received by the community [31, 38], also stimulated criticism about the appropriate ways to identify differentially expressed genes [3, 19, 31, 35, 38]. Disagreements among scientists should provide part of the energy and process to move to consensus on the "best practices" for the generation, analysis, and application of microarray data. This is precisely the goal of the MAQC project toward personalized medicine that is the future of healthcare.

References

1. Allison, D. B., et al. (2006). Microarray data analysis: From disarray to consolidation and consensus. *Native Reviews. Genetics*, *7*, 55–65.
2. Canales, R. D., et al. (2006). Evaluation of dna microarray results with quantitative gene expression platforms. *Nature Biotechnology*, *24*, 1115–1122.

3. Chen, J. J., et al. (2007). Reproducibility of microarray data: A further analysis of microarray quality control (MAQC) data. *BMC Bioinformatics*, *8*, 412.
4. Chen, L., et al. (2006). Mutations induced by carcinogenic doses of aristolochic acid in kidney of Big Blue transgenic rats. *Toxicology Letters*, *165*, 250–256.
5. Ein-Dor, L., et al. (2006). Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proceedings of the National Academy of Sciences of the United States of America*, *103*, 5923–5928.
6. Fodor, S. P., et al. (1991). Light-directed, spatially addressable parallel chemical synthesis. *Science*, *251*, 767–773.
7. Frantz, S. (2005). An array of problems. *Nature Reviews. Drug Discovery*, *4*, 362–363.
8. Frueh, F. W. (2006). Impact of microarray data quality on genomic data submissions to the fda. *Nature Biotechnology*, *24*, 1105–1107.
9. Geiss, G. K., et al. (2008). Direct multiplexed measurement of gene expression with color-coded probe pairs. *Nature Biotechnology*, *26*, 317–325.
10. Gunderson, K. L., et al. (2004). Decoding randomly ordered dna arrays. *Genome Research*, *14*, 870–877.
11. Guo, L., et al. (2006). Rat toxicogenomic study reveals analytical consistency across microarray platforms. *Nature Biotechnology*, *24*, 1162–1169.
12. Hoffman, E. (2004). Expression profiling–best practices for data generation and interpretation in clinical trials. *Native Reviews. Genetics*, *5*, 229–237.
13. Hughes, T. R., et al. (2001). Expression profiling using microarrays fabricated by an ink-jet oligonucleotide synthesizer. *Nature Biotechnology*, *19*, 342–347.
14. Ioannidis, J. P. (2005). Microarrays and molecular research: Noise discovery? *The Lancet*, *365*, 454–455.
15. Irizarry, R. A., et al. (2005). Multiple-laboratory comparison of microarray platforms. *Nature Methods*, *3*, 345–350.
16. Irizarry, R. A., et al. (2006). Comparison of Affymetrix GeneChip expression measures. *Bioinformatics*, *22*, 789–794.
17. Ivanova, N. B., et al. (2002). A stem cell molecular signature. *Science*, *298*, 601–604.
18. Kadota, K., et al. (2009). Ranking differentially expressed genes from affymetrix gene expression data: Methods with reproducibility, sensitivity, and specificity. *Algorithms for Molecular Biology*, *4*, 7.
19. Klebanov, L., et al. (2007). Statistical methods and microarray data. *Nature Biotechnology*, *25*, 25–26. Author reply 26–27.
20. Lockhart, D. J., et al. (1996). Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, *14*, 1675–1680.
21. Marshall, E. (2004). Getting the noise out of gene arrays. *Science*, *306*, 630–631.
22. Mecham, B. H., et al. (2004). Sequence-matched probes produce increased cross-platform consistency and more reproducible biological results in microarray-based gene expression measurements. *Nucleic Acids Research*, *32*, e74.
23. Mei, N., et al. (2004). Differential mutagenicity of riddelliine in liver endothelial and parenchymal cells of transgenic big blue rats. *Cancer Letters*, *215*, 151–158.
24. Mei, N., et al. (2004). Mutations induced by the carcinogenic pyrrolizidine alkaloid riddelliine in the liver cII gene of transgenic big blue rats. *Chemical Research in Toxicology*, *17*, 814–818.
25. Mei, N., et al. (2005). Mutagenicity of comfrey (Symphytum Officinale) in rat liver. *British Journal of Cancer*, *92*, 873–875.
26. Michiels, S., et al. (2005). Prediction of cancer outcome with microarrays: A multiple random validation strategy. *The Lancet*, *365*, 488–492.
27. Miklos, G. L., & Maleszka, R. (2004). Microarray reality checks in the context of a complex disease. *Nature Biotechnology*, *22*, 615–621.
28. Miller, R. M., et al. (2004). Dysregulation of gene expression in the 1-methyl-4-phenyl-1,2,3,6-tetrahydropyridine-lesioned mouse substantia nigra. *Journal of Neuroscience*, *24*, 7445–7454.
29. Ramakrishnan, R., et al. (2002). An assessment of Motorola CodeLink microarray performance for gene expression profiling applications. *Nucleic Acids Research*, *30*, e30.

30. Ramalho-Santos, M., et al. (2002). ‘stemness’: Transcriptional profiling of embryonic and adult stem cells. *Science*, *298*, 597–600.
31. Sage, L. (2006). Do microarrays measure up? *Analytical Chemistry*, *78*, 7358–7360.
32. Schena, M., et al. (1995). Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, *270*, 467–470.
33. Shi, L., et al. (2005). Cross-platform comparability of microarray technology: Intra-platform consistency and appropriate data analysis procedures are essential. *BMC Bioinformatics*, *6*(Suppl. 2), S12.
34. Shi, L., et al. (2006). The MicroArray Quality Control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. *Nature Biotechnology*, *24*, 1151–1161.
35. Shi, L., et al. (2007). Reply to Statistical methods and microarray data. *Nature Biotechnology*, *25*, 26–27.
36. Shi, L., et al. (2008). The current status of DNA microarrays. In Dill K., Liu R., & Grodzinski P. (Eds.), *Microarrays: Preparation, microfluidics, detection methods, and biological applications* (pp. 3–24). New York: Springer.
37. Shi, L., et al. (2008). The balance of reproducibility, sensitivity, and specificity of lists of differentially expressed genes in microarray studies. *BMC Bioinformatics*, *9*(Suppl. 9), S10.
38. Strauss, E. (2006). Arrays of hope. *Cell*, *127*, 657–659.
39. Su, Z., et al. (2009). Approaches and practical considerations for the analysis of toxicogenomics data. In Boverhof D.R., & Gollapudi B.B. (Eds.), *Application of toxicogenomics in safety evaluation and risk assessment*. Wiley, Chichester, West Sussex, UK.
40. Tan, P. K., et al. (2003). Evaluation of gene expression measurements from commercial microarray platforms. *Nucleic Acids Research*, *31*, 247–276.
41. Tusher, V. G., et al. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America*, *98*, 5116–5121.
42. Wang, E. T., et al. (2008). Alternative isoform regulation in human tissue transcriptomes. *Nature*, *456*, 470–476.
43. Wang, Y., et al. (2006). Large scale real-time PCR validation on gene expression measurements from two commercial long-oligonucleotide microarrays. *BMC Genomics*, *7*, 59.

Chapter 10
A Survey of Classification Techniques for Microarray Data Analysis

Wai-Ki Yip, Samir B. Amin, and Cheng Li

Abstract With the recent advance of biomedical technology, a lot of 'OMIC' data from genomic, transcriptomic, and proteomic domain can now be collected quickly and cheaply. One such technology is the microarray technology which allows researchers to gather information on expressions of thousands of genes all at the same time. With the large amount of data, a new problem surfaces – how to extract useful information from them.

Data mining and machine learning techniques have been applied in many computer applications for some time. It would be natural to use some of these techniques to assist in drawing inference from the volume of information gathered through microarray experiments.

This chapter is a survey of common classification techniques and related methods to increase their accuracies for microarray analysis based on data mining methodology. Publicly available datasets are used to evaluate their performance.

10.1 Summary and Outline

Microarray is a new and important technology in exploring certain kinds of biological data. This chapter provides some of the background in bioinformatics and the technology behind microarray chip. It also provides the motivations and challenges

C. Li (✉)
Dana Farber Cancer Institute, 44 Binney Street, CLSB 11036, Boston, Massachusetts 02115
e-mail: cli@hsph.harvard.edu

W.-K. Yip
Harvard School of Public Health, 677 Huntington Avenue, Boston, Massachusetts 02115
e-mail: wkyip@hsph.harvard.edu

S.B. Amin
Dana Farber Cancer Institute, 44 Binney Street, Boston, Massachusetts 02115
e-mail: amin@jimmy.harvard.edu

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_10,
© Springer-Verlag Berlin Heidelberg 2011

behind the methodology used to explore the vast amount of data generated using microarray.

This chapter focuses on one particular area – supervised learning or classification. In general, microarray prediction is a multi-class classification problem. In order to simplify complexity, we narrow our discussion only to 2-class. Six of the commonly used classification techniques are described here: Decision Tree, k-Nearest-Neighbor, Discriminant Analysis, naive Bayesian classifiers, Support Vector Machine and Artificial Neural Network. For each of the method, we provide a basic description of the theory, a brief summary of the method's advantages and disadvantages, and a short survey of recent research in the area. Accuracy as a measurement metric is introduced followed by descriptions of techniques used for improving accuracy, such as Bagging and Boosting. Various validation techniques such as Cross Validation and Bootstrap, are discussed. Real data sets are used as examples for evaluating the classification techniques.

A number of software packages are now available to analyze microarray data sets and five such packages are described here – BRB-Array Tools, Bioconductor, GenePattern, PAM, and dChip.

Finally, a discussion about remaining issues and challenges is presented.

10.2 The Bioinformatic Revolution – The Challenges of the Biomedical Data

Bioinformatics is a new area of research that applies mathematics, computer information technology to the field of molecular biology. It helps to solve any theoretical and practical problems that arises from analyzing biological data. It includes special data management techniques to manage large amount of data generated, specialized computational algorithms to find answers quickly, and statistical methods to analyze the data appropriately.

For the last few decades, advance and rapid developments in both molecular biology and computer technologies led to the generation of a large amount of information. The most notable example is the Human Genome Project [1]. With its completion in 2003, it has identified all of the approximately 20,000–25,000 genes in human, determined the sequences of the three billion chemical base pairs that make up human DNA, and stored this information in databases which is now available for all researchers.

With the explosion of genomic data, data mining (also known as knowledge discovery) techniques have been applied to "mine knowledge" from these data. The knowledge gleaned from these data could be invaluable in understanding the underlying biological processes and so it can help us to diagnose more precisely medical problems such as cancer, and to predict the outcome of various therapies [2].

10.3 DNA Microarray Technology and Data Analysis

DNA microarray technology is developed to allow researchers to collect a very large number of gene expressions at the same time. Genes are expressed under different conditions and times and made into proteins. The instructions are transcribed from the DNA in the genes that reside in the nucleus by messenger RNA (mRNA) and the actual assembly of the protein occurs in the ribosome of the cell. As a result, the state of the cell is correlated with changes in the level of mRNAs. By measuring the level of mRNA, the state of the cell can be determined and inference about what is happening in the cell can be made [3].

DNA microarray is a chip made of silicon or glass as in Affymetrix array or microscopic beads as in Illumina array where thousands of strands of complimentary DNA (cDNA) molecules or oglinucleotides are implanted and lay out in a grid fashion. A mixture of mRNAs derived from the cells are then allowed to hybridize (bind) to the probe sequences on the chip. The level of hybridization is detected through fluorescence dyes with imaging software from chip manufacturers. It is expected that the concentration level of each mRNA is proportional to the intensity of hybridization detected. Researchers can now investigate the expressions of thousands of genes all at the same time. By comparing the results against a control (normal cell), the researcher can now quantify changes in gene expression levels.

Since researchers want to have a global view of gene expression, a large number of genes are usually included for microarray experiment. It introduces a challenge that is unique to microarray gene expression data – a large number of gene expressions (in thousands) with a relatively small number of samples (at most several hundreds). There may be more “false positives” happening due to chance. Thus robust methods for validating the models and evaluating their accuracies are needed [4].

Microarray data are usually presented as a heatmap. An example is shown in Fig. 10.1. Due to the large number of genes involved, the array is arranged with samples in the column and genes in the rows. Both samples and genes are arranged according to an hierarchical clustering method. The red indicates high gene expression level while the blue indicates low gene expression level. The heatmap gives a visual summary of genetic profiles from samples.

There are many applications of microarray experiments such as establishing genome-wide DNA methylation maps and measure DNA copy numbers across the genome. The most prominent application is in cancer research. In 1999, a group of scientists led by Golub published the pioneer work using gene expression to classify cancer [6]. With DNA samples from leukemia patients, 50 genes were selected from a total of 6817 genes as informative genes. Based on the data, the computer program is trained to classify (supervised learning) two kinds of leukemias – acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). Then classification of new leukemia cases was performed using individual gene selection and a voting algorithm. The study also applied an unsupervised learning technique, self organizing maps (SOMs), to discover the distinction between AML and ALL, as well as the distinction between B-cell and T-cell ALL.

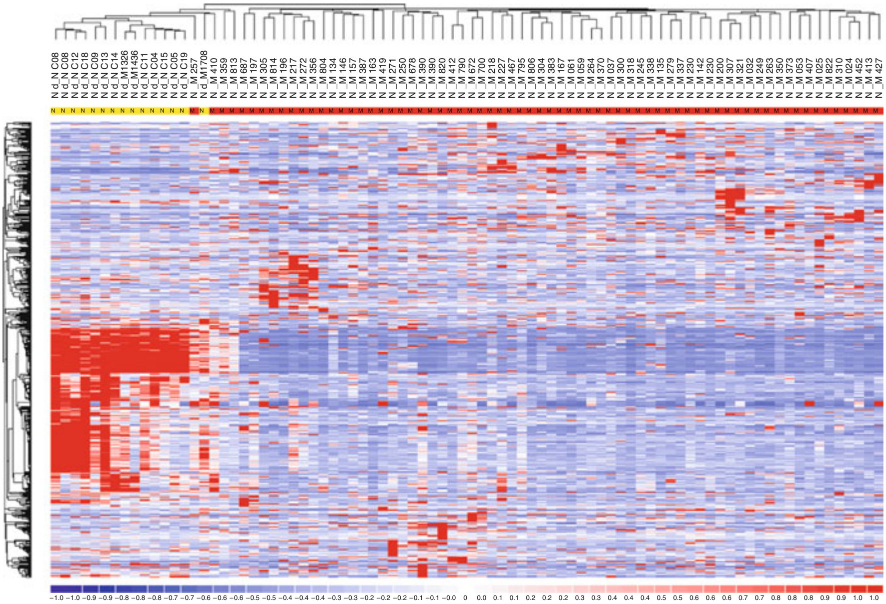


Fig. 10.1 The heatmap figure is showing unsupervised clustering of 15 healthy donor plasma cell samples (*yellow*) and 69 new myeloma samples (*red*) from the NCBI GEO dataset GSE6477 [5]. Following data normalization, gene filtering by dChip was carried out using variance (SD/mean) range between 1.4 and 1000 and minimum probeset level expression of 20 in at least 50% of samples to get 460 genes. $1-$*correlation* cluster algorithm was used using average linkage method with gene-level and sample-level function enrichment p-values of 0.001 and 0.01 respectively

A similar approach by Shipp et al. was published in 2002 [7]. The same technique was applied first to train the classifier to distinguish Diffuse Large B-Cell Lymphoma (DLBCL) treatment outcome. The derived class predictor was also able to predict if a patient with DLBCL will be cured or not.

These and other subsequent studies demonstrate the viability of using gene expression for developing models for cancer classifications and discovery. The importance of Microarray Analysis can be visualized by looking at the trend of the number of publications in major biomedical journals for the last 10 years as shown in Fig. 10.2.

In general, there are two kinds of machine learning: unsupervised learning (clustering, class discovery) and supervised learning (classification). In unsupervsed learning, samples are allowed to group together according to some criteria without any guidance. New classes may be discovered by looking at how the samples are grouped together naturally. Self Organizing Maps (SOM), Principal Component Analysis (PCA) or Hierarchical Clustering are such techniques. There are quite a lot of research in that area as well and will not be discussed here. Instead, this chapter focuses on supervised learning when samples with known outcomes are used to

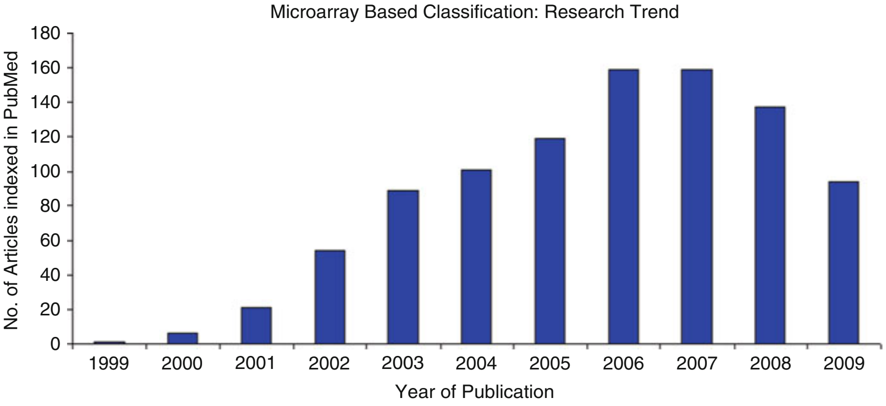


Fig. 10.2 The graph shows the number of articles cited in Index Medicus related to microarray based classification methods in the last 10 years. The result is obtained by searching the PubMed database for articles published between 1999 and 2009 with titles containing the word 'microarray'

teach the classifer. Once the training or teaching is done, the classifer can be used to predict the outcome on future input data.

After collecting gene expression data, the first step is to preprocessing the data so that they are ready for analysis. There are many data preprocessing techniques and a good description can be found in [2]. Two extremely important ones are data imputation for missing data and normalization which will discuss briefly here.

In microarray experiments, fluorescent intensities, which are related to gene expression levels, are measured. Comparing the intensities directly may be biased because there could be systematic variations due to scanning parameters settings and differentials in dyes usage. Normalization can be used to eliminate these systematic effects so that the data after normalization reflect the true differentials in gene's expression levels. A simple normalization procedure transforms the data so that each gene expression has mean zero and variance of one across. By doing so, all the genes are weighted equally in the classification. The location and scale parameters can be estimated by using the mean and standard deviation of the sample. A more robust estimator such as the median can also be used. Some authors recommends general adaptive and robust normalization procedures such as the robust local regression to correct for some of these artifacts [2].

The next step is to do data analysis which may include the following steps [7]:

1. Class definition – define a class label based on morphology, tumor class or some treatment outcome information and this is usually done before data analysis and during data collection;
2. Feature selection – select a set of non-redundant genes that are most relevant (or highly correlated) to the classes in question based on some separation statistics;
3. Classification – build, train and validate a classifier using existing data set based on the classification; then use the classifier to predict outcomes based on gene expression;

4. Model selection – build several models using different sets of features and chose the final model based on some criteria to minimize the total error in cross-validation;
5. Model evaluation – evaluate prediction results, and apply other measures such as ROC curves or Kaplan-Meier survival plots for evaluation;
6. Cluster – identify new biological disease cases or make refinement on existing classification.

Note that this is idealization of the process. Some of the steps may not be always separable. In fact, part 4 and 5 are usually included in part 3 for model selection and evaluation. For example, classifiers such as support vector machines have feature selection built directly together with the training process. After validation, the model is applied to an independent set of test data for an unbaised assessment. Classifiers such as support vector machine have feature selection built directly together with the training process.

10.4 Classification Techniques for Microarray Analysis

This chapter discusses only classification techniques and related methods to increase the accuracies. The main goal is to come up with a 'reliable' classifer that can be used to predict the sample classes based on gene expression data.

This section describes common classification methods. This is by no means an exhaustive list. Reader can learn more about machine learning techniques in the classic textbook *Data Mining: Concepts and Techniques* by Jiawei Han and Micheline Kamber [8] or the online lecture notes by Professor Andrew Moore [9]. This chapter will cover six classification methods that are most commonly used and have implementations in stable software packages such as R and those described in Sect. 10.8. The simpler ones are presented first and are followed by more sophisticated ones. Here are the six classification methods:

1. Decision Trees
2. K-Nearest Neighbor
3. Discriminant Analysis
4. Support Vector Machines
5. Artificial Neural Network
6. Naive Bayesian Classifiers/Bayesian Network

Theoretically, it can be shown with simplified assumptions, the classification techniques are different schemes of finding the class label that maximize the likelihood of data for the frequentist approach or the posterior probability for the Bayesian approach given the observation.

The following general notation is used. It is assumed that data is presented as a set of tuples (O, L) where O is an observation comprising of many attributes (genes) at once and L is the class label (e.g., disease outcome). Sometimes, in order to reduce complexity, only a subset of attributes, F, is considered. The process of selecting the attribute subset F from O is called feature selection. For microarray analysis, O is usually expression level from many different genes, F is a subset of O called the feature set, and L is the disease or treatment outcome. Training set, T, refers to the set of tuples used to train the classifier and the testing set, V, refers to the set of tuples used to validate the learned classifer. The count of the number of elements in the set O is denoted by $|O|$.

10.4.1 Decision Trees

Decisions can easily be represented in a form of a tree. In its basic form, a decision tree is a tree structure comprising internal nodes and leaves where each internal node denotes a test on an attribute or integration of several attributes; each branch represents an outcome of the test; and each leaf node holds a class label.

The decision tree is being built by learning from the training set T. Once the decision tree is built, it gives you the knowledge repository for prediction for any future input. Since the construction of some decision tree classifiers require only limited knowledge of the subject matter or setting any parameters, therefore, it is appropriate for exploratory knowledge discovery. In fact, it is like a game of 20 questions. It generates a model that is both predictive and descriptive. Since the knowledge is encoded in a tree structure, one can query the tree for how exactly it arrives at every decision. An example of a decision tree is shown in Fig. 10.3 which is the result of learning from Shipp's DLBCL data set.

Statisticians Breiman, Friedman, Olshen and Stone and a machine learning researcher, Ross Quinlan independently developed the algorithm. The technique is also known as Classification and Regression Tree (CART). An implementation is available in R under the **rpart** package. Reader can learn more by reading the textbook *Classification and Regression Trees* by Breiman et al. [10].

The following is a simplified recursive algorithm for binary attribute selection to generate a decision tree:

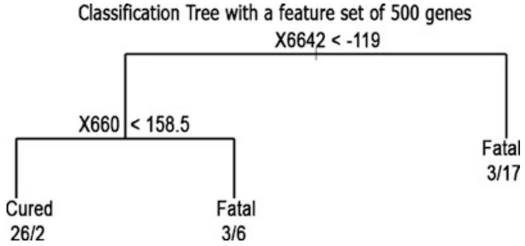


Fig. 10.3 The decision tree to predict cancer outcome using 500 genes as the feature set from Shipp's DLBCL dataset. The algorithm selected just 2 genes out of 500 as predictors. The number of observations for cured/fatal are shown beneath each leaf node

Input:

1. T, set of training tuples with observations and their associated class labels
2. A, set of candidate attributes
3. an attribute selection method

Output: A decision Tree
Method: generate_decision_tree (T, A)

1. create a node N;
2. if tuples in T are all with the same class label, C, then return N as a leaf node labeled with the class label C;
3. if all attributes in the attribute list are used, then return N as a leaf node labeled with the majority class label in T;
4. apply attribute selection method to find the 'best' splitting criterion based on T;
5. label node N with splitting criterion;
6. mark the attribute in A used for splitting;
7. for each outcome j of splitting criterion let T_j be the set of data tuples in T satisfying outcome j; if T_j is empty then attach a leaf labeled with the majority class in T to node N else **recursively** apply the algorithm to the subset T_j and attach the node return by *generate_decision_tree*(T_j, A) to node N;
8. return N;

Different attribute selection methods can be applied to decide how to pick the best split for the attribute set. Let D be a subset of T of learning tuples being evaluated. Let p_i be the probability that a tuple in D that has class label l and is estimated by number of tuples in class l divided by total number of tuples in D. The following are some of the common measures for binary attributes:

1. Information gain – use the entropy function, $\mathit{info}(D) = -\sum p_i log_2(p_i)$ as the criteria. Since the decision based on A is binary, the set D can be partitioned into two sets D_i and D_j. Define $\mathit{Info}(D, A) = |D_i|/|D| \times \mathit{Info}(D_i) + |D_j|/|D| \times \mathit{Info}(D_j)$. Information gain is $\mathit{Info}(D) - \mathit{Info}(D, A)$. Pick A such that the information gain is largest.
2. Gini index – uses the gini index measure defined as $1 - \sum(p_i)^2$. Since the decision based on A is binary, the set D can be partitioned into two sets D_i and D_j. Define $\mathit{Gini}(D, A) = |D_i|/|D| \times \mathit{Gini}(D_i) + |D_j|/|D| \times \mathit{Gini}(D_j)$. Pick the attribute, A, as the splitting criteria such that $\mathit{Gini}(D) - \mathit{Gini}(D, A)$ is largest.

The advantages of using decision tree are

1. it is simple to use; and
2. it is easily understandable as it explains how it arrives at its conclusion.

The disadvantages are

1. it usually overfits the data;
2. it does not compare favorably with other machine learning techniques in terms of accuracy; and

3. it is hard to grow and reorganize the tree with new information. Overfitting the data is a problem as the algorithm tries to fit all data including noises as well. Tree pruning, which will not be described here, are techniques to remove the least reliable branches.

Since accuracy is a problem, regression trees classification technique is used commonly together with accuracy boosting techniques (described in later section). Random Forest, Adaboost are just some of the ensemble techniques. With accuracy boosting ensemble techniques and tree pruning improvements, microarray data can now be analyzed with high accruacy. Subject matter knowledge can be applied to prune the trees to avoid overfitting. It is the only techniques discuss here that can explain how it arrives at the prediction.

Early work to apply decision trees technique for cell and tumor classification using gene expression data was done by Zhang et al. [11]. They introduced a deterministic procedure to form forests of classification trees. Their performance in terms of error rates were compared with alternative techniques.

Recent research claims to produce tree based ensemble methods that produce highly accurate results for microarray data by using Partial Least-Squares (PLS) regression as a feature selection method [12]. The paper suggests a two stage dimensionality reduction scheme:

1. Removal of irrelevant genes using discretization method;
2. Feature selection using PLS.

The number of features are selected through the SIMPLS algorithm, an alternative estimation method for partial least squares regression components proposed by de Jong. For validation, it uses both ten-fold as well as leave-one-out-cross validations. The results using four different decision tree methods: Simple C4.5, Random Forest, C5.0 Adaboost, and MML(Minimum Message Length) Oblique Forest are produced. Seven publicly available cancer data sets, Leukemia, Breast cancer, Central nervous system, Colon tumor, Lung cancer, Prostate cancer, and Prostate cancer, are used. It shows that the Partial Least-Squares (PLS) regression method is an appropriate feature selection method and tree-based ensemble models can deliver accurate classification models for microarray data.

Recently, a successful optimization technique is proposed for decision tree classifiers by using hist index to prune attributes, approximating computations that measure entropy, and reusing subtrees from previous runs [13]. The algorithm is applied to three public cancer data sets – leukemia (from Golub), colon and breast. In all cases, the CPU consumption drops significantly. In addition, the paper also noticed that smaller tree seems to have higher accuracy rate as well.

10.4.2 k-NN (k-Nearest-Neighbor)

The k-NN method is a classification algorithm by gathering information from its neighbors. It belongs to a class of learning techniques known as lazy learner [8].

A lazy learner simply stores the training tuples with minimal processing. When the test tuple is presented, the learner examines and then classifies the test tuple based on its similarity to the stored training tuples. The most common class label among the k nearest neighbors to the test tuple. This type of classifer is also known as instance-based learner because of the way it stores all the instances of the training set. This technique is inherently computationally intensive as most of the work is done at the time of classification.

The k-nearest-neighbor method has been around since the early 1950s. Because of its heavy computational requirement, the method is not actively in use until computers are widely available. It classifies by comparing a given test tuple with training tuples that are similar to it where similarity (or closest) is defined by some distance measure – usually the Euclidean distance. The class label that is most common to the test tuple's k nearest training tuples is assigned. There are no specific rules to choose what is the best k. This is usually chosen by the intuition of the researchers. Missing values and categorical variables can be handled easily by assuming maximum possible differences.

An implementation of k-NN algorithm in R can be found in the **class** package.

The following are a list of common distance functions used [2]. Let x and y are members (genes) of the feature set, F, and x_i and y_i are the i-th components of x and y respectively.:

1. Euclidean – $dist(x, y) = \sqrt{\sum (x_i - y_i)^2}$
2. Manhattan – $dist(x, y) = \sum |x_i - y_i|$
3. Mahalanobis – $dist(x, y) = \sqrt{(x_i - y_i)S^{-1}(x_i - y_i)^T}$ where S is the sample covariance matrix of the variables.

The advantages of using k-NN are

1. it is simple to use;
2. it is easily understandable;
3. it allows better control as user can pick an appropriate distance function; and
4. it can incorporate new information easily.

The disadvantages are

1. it is not as accurate in prediction as other complicated machine learning; and
2. it is computationally intensive.

Since it is easy to use and understand and there is no prior knowledge needed of the data, k-NN has been used frequently as a comparison to other machine learning techniques when applied to microarray data. Surprisingly, it works pretty well although not as good, but not far from the best score when compared with the more sophisticated methods.

As presented earlier, the k-NN technique was applied early on by Margaret Shipp on her tumor classification work [7]. k-NN was used to predict the outcome of diffuse large B-cell lymphoma. The results was used to compared with other techniques such as SVM.

Recent research involves exploring feature standardization and fuzzification to improve accuracies based on receiver operating characteristic (ROC) curves [14].

10.4.3 Discriminant (linear/quadratic)

Linear discriminant analysis (LDA) is a statistical and machine learning method by finding the best linear combination of features which separate two or more classes of objects when each measurement of the covariates is continuous. The combination can then be used as a classifier, or, more commonly, for dimensionality reduction before classification in the second stage.

Discriminant function analysis is based on the assumption that the variable for each group is normally distributed within each class k with mean $= \mu_k$ and standard deviation $= S_k$. And the joint distribution is a multivariate normal distribution $N(\mu_k, S_k)$ where μ_k denotes the expected value and S_k denotes the covariance matrix of the feature in class k. Take the case when k = 2, the Bayes optimal solution is to predict points as being from the second class if the likelihood ratio is below some threshold T, i.e.:

$$((x - \mu_1)S_1^{-1}(x - \mu_1)' + log|S_1|) - ((x - \mu_2)S_2^{-1}(x - \mu_2)' + log|S_2|) < T$$

This classifier is known as the quadratic discrminant analysis (QDA). If further assumptiion is made that all the standard deviations are the same, the criteria can be reduced to a linear combination of just the observations. The resulting classifier is known as the linear discriminant analysis (LDA). Based on assumptions on the variance-covariance matrices of the variables, there are variants of the discriminant analysis: diagonal quadratic discriminant analysis (DQDA) – when all the class densities have their variance-covariance matrices as diagonal matrices, diagonal linear discriminant analysis (DLDA) – when the all class densities variance-covariance matrices are the same diagonal matrix [2].

The advantages are:

1. it is a well-known statistical technique that has been proven to work over the years;
2. it is easy to implement (because of its linear nature); and
3. it has good performance in practice (high bias, but low variance leads to good estimates.

The disadvantages are:

1. it assumes that the underlying model is multivariate normal distribution; and
2. the traditional LDA may not be suitable for microarray analysis because for a large number of genes as performance degrades rapidly due to over-parameterization and high variance parameter estimators.

An implementation of **lda** and **qda** can be found in library **MASS** in R.

As presented earlier, Golub et al. [6] proposed a weighted gene voting scheme for classification. This method is a variant of DLDA and is considered as one of the first applications of a classification method to gene expression data [2].

The technique is very flexible and has been extended to deal with some of its shortingcomings. LDA can be adapted to Microarray data easily with various shrinkage approaches. The newer extensions (see below) can easily handle high dimensiion data with small sample size common in Microarray experiments. The following are two such methods from recent research:

A new variant of LDA, sequential DLDA (SeqDLDA), has been proposed. It is based on Diagonal LDA (DLDA) combined with an independent gene selection (filtering) – one gene is sequentially added each time and the linear discriminant (LD) recomputed using the DLDA model at each iteration. Classical DLDA adds the gene with highest t-test score without checking the resulting model. SeqDLDA improves on the method by finding the gene that better improves class separation after recomputing the model measured using a robustified t-test score. The new method was used in several 2-class cancer datasets (neuroblastoma, prostate, leukemia, colon) using ten-fold cross-validation. The misclassification rate is significantly reduced in some of the data set [15].

The shrunken centroids regularized discriminant analysis (SCRDA) is another new variant [16]. The method generalizes the idea of the nearest shrunken centroids of Prediction Analysis of Microarray (PAM) into the classical discriminant analysis. It is specially suited for microarray data as it is designed for classification problems in high dimension data with low sample size. Using both simulation study and real life data, SCRDA perform uniformly well in the multivariate classification problems.

10.4.4 Support Vector Machines

Support vector machines (SVM) is one of the most promising sophisticated machine learning techniques. The technique was first proposed by Vladimir Vapnik and Alexei Chervonenkis in 1970s [17].

The idea can best be explained in a simple case when the data are linearly separable, i.e., a line (or a hyperplane) that can be drawn to clearly separate the data into two classes. The goal of the algorithm is to find that line (or hyperplane). Constraints are based on margin which is the shortest distance of vectors to this line (or hyperplane). The shortest distance from vectors of one class to the dividing line (or hyperplane) should be the same as the short distance from vectors of another class. The maximum marginal hyperplane is the hyperplane that has the maximum margin.

Not all data are linearly separable. That is, a line (or a hyperplane) cannot be drawn through the data such that one class of data is completely on one side and the other class is on the other. In those cases, a kernel function can be used to transform the data to a higher dimension so that the transformed data points in the higher dimension can be linearly separable. Kernel function is being applied cleverly to the algorithm to avoid computational burden.

The following are commonly used kernel functions:

1. Polynomial of degree $p - (x \cdot y + 1)p$
2. Gaussian radial basis (RBF) – $exp(-|x - y|^2/2\sigma^2)$
3. Sigmoidal – $tan^{-1}(\kappa x \cdot y - d)$

By default, a linear polynomial function is used. In addition, user can also specify the more advance classification parameter – the C-classification, the ν-classification, or the one-classification.

The advantages are:

1. it is highly accurate;
2. it is less prone to overfitting;
3. it can model very complex, nonlinear data; and
4. it provides a compact description of the learned model.

The disadvantages are:

1. learning can be computational intensive; and
2. it takes skills to pick the right kernel functions for the right problems with the right parameters.

An implementation of SVM can be found in the library **e1071** in R. Due to its generality, SVM has been applied to many areas. Since SVM can easily deal with complex high dimension data, it has been applied successfully to microarray analysis. It is a very active area as researchers are trying to come up with proper methodology to pick the right classification parameter and the appropriate kernel function.

Michael Brown et al. were the first to publish an analysis of microarray gene expression data by using support vector machines [18]. It stated many mathematical features about SVMs that make them attractive for gene expression analysis, including their flexibility in choosing a similarity function, sparseness of solution when dealing with large data sets, the ability to handle large feature spaces, and the ability to identify outliers. Several SVMs were tested, as well as some other supervised learning methods, and find that the SVMs best identify sets of genes with a common function using expression data.

In [19], the paper proposes a gene selection method based on RFE (recursive feature elimination). This becomes the dominant SVM feature selection method. Experimentally, the researchers demonstrated that the genes selected by the techniques yield better classification performance and are biologically relevant to cancer.

In [20], the paper proposes a variation of SVM – R-SVM (recursive SVM) to perform feature selection and classifcation with noisy data. Using simulated data, the paper claims to have 5–20% improvement over SVM-RFE (recursive feature elimination, the predominant SVM method). The method is being applied to two proteomic datasets – human breast cancer and mouse liver cirrhosis. It demonstrates the viability of the method for proteomic and microarray analysis especially when the data is noisy.

In [21], the paper proposed a variation of SVM – SCADSVM which performs feature selection and classification simultaneous. It is applied to three breast cancer data sets (Stanford, Rosetta, Singapore). The result is lower error rate with cross validation when compare with other SVMs and t-statistics. It uses ten-fold to tune the SVM parameter and then it uses two fold to train and one fold to test for the classifers. (i.e., train on Stanford and Rosetta, test on Singapore).

In [22], the paper proposes an extension to SVM-RFE (recursive feature elimination) to do multiclass classification by simultaneously considering all classes during the gene selection stages. A total of six data sets are used – all with multiple (>2) classes. Cross validation shows that the proposed extensions work well.

10.4.5 Neural Networks

Neural networks were originally proposed by psychologists and neurologists who tried to develop and test computational analogues of neurons. Like neurons in a human being, a neural network is a set of connected input/output units. To model an actual neuron, each connection has a weight associated with it. When the threshold in one this unit is reach, the connection is fired and the signal propagates to the next unit. The network learns by adjusting the weights so that the correct class label of the input tuples is obtained.

There are many kinds of neural networks and neural network algorithms. The simplest kind is the multi-layer feed-forward neural network which contains an input layer, one or more hidden layers, and one output layer. The signal can only propagate forward from the input layer through the hidden layers to the output layer. Backpropagation is the most commonly used algorithm to train this kind of neural network.

The most important aspect of neural network is the network topology which is supplied by the user. To set up a multi-layer feed-forward network topology, the user needs to specify the number of units in each layers (input, hidden and output), the number of hidden layers, and the connections of all the units. Input may need to be normalized. There is no rule to pick the best parameters. It is an iterative process of trial and error to find the best neural network.

Once the neural network is set up, the user can supply training tuples. Backpropagation learns by iteratively processing the training data set by comparing the network's prediction for each tuple with the actual known target value. For each training tuple, the weights of each unit are modified so as to minimize the mean squared error between the network's prediction and the actual target value. These modifications are made in the backwards direction, that is, from the output layer, through each hidden layer back up to the first hidden layer. The weights of each unit should eventually converge although there is no guarantee theoretically. The learning process stops then.

An implementation of neural network in R can be found in the **neural** package. A single single-hidden-layer neural network implementation is also available in the **nnet** package which comes with the base R.

The advantages are:

1. it is highly tolerant of noisy data;
2. it is capability of classifying patterns on which they have not been trained;
3. it is well-suited for continuous-valued data; and
4. it is successful on a wide range of real-world data (including microarray datasets.)

The disadvantages are:

1. it involves long learning time;
2. it requires a number of parameters that are typically best determined empirically; and
3. it is difficult to interpret the symbolic meaning behind the learned weights and hidden units in the network.

Since microarray data usually carries a lot of noise, ANN seems to be the right classifier. To set up the neural network properly, prior knowledge of the problem must be used. Experiments are conducted to obtain some of the weights used in the network units. So, the difficulties in training the neural network and in interpreting the inherent meaning of internal parameters make it hard to use in analyzing Microarray data. The first application of neural networks to microarray data classificiation is published by J. Khan et al. [23]. After the ANN was trained using the small, round blue-cell tumors (SRBCTs) as a model, it correctly classified all samples and identified the genes most relevant to the classification. The experiment demonstrated the potential for using these methods for tumor diagnosis and the identification of candidate targets for therapy.

The following are some of the recent research neural network:

In [24], the paper explores using Artificial Neural Network (ANN) trained on microarray data from DLBCL lymphoma patients to predict long term survival for cancer patients. The resulting classifier has been able to predict the long-term survival of individual patients with 100% accuracy. The paper concludes that artificial neural networks are a superior tool for analyzing microarray.

Also, it has been reported that the researchers at the National Cancer Institute (NCI), have used artificial neural networks (ANNs) and DNA microarrays to successfully predict the clinical outcome of patients diagnosed with neuroblastoma (NB) [25]. Out of 25,000 genes, the ANN classifier identified a minimal set of 19 genes whose expression levels were closely associated with this clinical outcome.

10.4.6 Naive Bayesian/Bayesian Network

Bayesian classifiers are based on Bayes statistical theory. They are used to compute class membership probabilities, that is the probability that a given tuple belongs to a particular class.

The foundation of Bayesian classification is based on the famous Bayes theorem. Its accuracy and speed are quite extraordinary when applied to large data sets. Bayesian classifiers are useful because they actually provide theoretical justification for many other classifiers if the theory is set in the Bayesian context. It can be shown that many machine learning algorithms actually output the maximum *a posteriori* hypothesis.

Bayes Theorem

$$P(H|X) = \frac{P(X|H) \times P(H)}{P(X)}.$$

where

$P(H|X)$ is the *a posteriori* probability of H conditioned on X.

X are known observations and $P(H)$ is the prior probability.

$P(X|H)$ and $P(H)$ are usually estimated from the given data.

Suppose that there are m classes, $C_1, C_2, \ldots C_m$. Given a tuple, $X = (x_1, x_2, \ldots, x_n)$, the classifier predicts that X belongs to the class having the highest *a posterior* probability, conditioned on X.

$$P(C_i|X) = \frac{P(X|C_i) \times P(C_i)}{P(X)}.$$

The objective then is to maximize $P(X|C_i)$ because $P(X)$ is fixed. Since the *prior* probability for C_i's, i.e., $P(C_i)$ s are not known, they are assumed to be identical. For the Naive Bayesian Classification, further assumption is made that the class conditionals are independent (i.e., given the class label of a tuple, the values of the attributes are assumed to be conditionally independent of one another). So, $P(X|H)$ is calculated by simply multiplying all the conditionals together, i.e., $P(X|H) = P(x_1|C_i) \times \cdots \times P(x_n|C_i)$ where all the $P(x_i|C_i)$ can be estimated from the training tuples. For prediction when given a tuple X, we just need to find the class C_j such that $P(X|C_j)P(C_j)$ is the maximum with the values that are estimated from training. If the class conditional independence assumption is not true, a more complicated method, Bayesian belief networks, can be used. The network specifies the joint conditional probability distributions. A belief network is represented by two data structures – a directed acyclic graph (DAG) and a set of conditional probability tables (CPT). Each node in the DAG represents a random variable that may correspond to the actual attributes given in the data or to some "hidden attributes" believed to form a relationship. If nodes are connected by an arc, it means that they are probabilistic dependent. The CPT at each node, Y, enumerates the conditional distribution $P(Y|Parents(Y))$.

The network provides a complete representation of the existing joint probability distribution by the formula [8]:

$$P(X) = \prod P(x_i|Parents(Y_i))$$

for all i and Y_i is the variable at each node.

The network is either given or learnt from the training set. The training algorithm is far more complicated than the naive Bayesian classifier and is beyond the scope of this chapter. Each node in the network is a class label although some of them may be "hidden". With the network, we can find the probability of each class based on the input tuple, X. We classify the input tuple to the class that has the highest posteriori probability.

There is no package in R for a naive Bayesian classifier. An implementation of Bayesian network can be found in package **deal** in R.

The advantages are:

1. its representational power as it gives the actual probabilities of the classification based on the input; and
2. it provides theoretical basis for other classifiers.

The disadvantages are:

1. it is difficult to train the network;
2. it is difficult to explain especially the hidden variables; and
3. it could be computational intensive (Bayesian network).

Bayesian network is the only technique here that models not only the relationship between input and class labels but also the joint distribution of the class labels and input. As a result, it can successfully predict outcome even when only partial information is available. However, the conditional probabilities are not trivial to compute. The interpretation of the network is not obvious. So, it is difficult to apply Bayesian network for Microarray analysis. Early work of applying Bayesian networks to analyze expression data was done by Friedman [26]. A framework built on the use of Bayesian networks for representing statistical dependencies was used for discovering genes interactions. A method for recovering gene interactions from microarray data was described.

The following are some recent publications applying Naive Bayesian classifier:

In [27], the paper applies the naive Bayesian classifier to classify genes for housekeeping or tissue specific for human, mouse or fruit fly based only on physical and functional characteristics of genes already available in databases, like exon length and measures of chromatin compactness. The classifier has achieved a 97% success rate in classification of human housekeeping genes (93% for mouse and 90% for fruit fly). The result is validated with a ten-fold random sampling cross validation.

In [28], the paper develop techniques that address the complexities of learning Bayesian nets. It reduces the Bayesian network learning problem to the problem of learning multiple subnetworks, each consisting of a class label node and its set of parent genes. This model is more appropriate for the gene expression domain than are other structurally similar Bayesian network classification models. Two other significant contributions are the construction of classifiers from multiple, competing Bayesian network hypotheses and algorithmic methods for normalizing and binning gene expression data in the absence of prior expert knowledge. The classifiers are validated on out-of-sample test sets and attain a classification rate in excess of 90% for two publicly available datasets. The results are comparable to, or better than, other classification methods.

In [29], the paper extends a Naive Bayes Model to take into account within sample heterogeneity. It demonstrates that explicitly dealing with heterogeneity can improve classification accuracy on a TMA prostate cancer dataset. The approach is validated by simulated data and the TMA dataset by applying 100 times a ten fold cross-validation procedure with different fold randomization and then by computing the average results.

10.5 Accuracy Enhancements to Classification

The accuracy of some of the classification techniques can be improved with two general techniques: bagging and boosting. Both of them are called ensemble methods as they base the result on a combination of the results from different models. The final result is a composite model usually has better overall classification/prediction accuracy.

10.5.1 Bagging

Bagging is another term for bootstrap aggregation [30]. It deploys bootstrapping procedures repetitively to come up with different training tuples to be used in training.

The user specifies the number of models needed in the ensemble (say, n). The bootstrap procedures are applied to resample with replacement the training tuples. Since it is resampling with replacement, some training tuples may be duplicated. The training process is being applied to the bootstrapped training set to come up with a model. The process is repeated n times. The end result is n different models.

Random Forest is an example for using such scheme. It is an ensemble of tree classifiers where bagging is used to produce the training set. Then, a random feature set is chosen to decide how to split the tree.

The composite model for classification is applying all n models to the input tuple. Each model votes with equal weights and the majority wins.

The bagged classifier often has significantly greater accuracy than a single classifier. It is more robust to noise. Application of bagging to cluster analysis can substantially improve clustering accuracy and yields information on the accuracy of cluster assignments for individual observations. In addition, bagged clustering procedures are more robust to the variable selection scheme, i.e., their accuracy is less sensitive to the number and type of variables used in the clustering [31,32].

10.5.2 Boosting

Similar to bagging, boosting also deploy boostrapping procedures to come up with n different models. However, it includes a modification in the training step. Each training tuple is assigned a weight. Initially, the weights are the same for each tuple. Then, it increases or decreases base on whether it was misclassified during the training process for the previous model. For classification, each model's prediction on the outcome when presented with an input is weighted according to how well it performs. The weight is usually set to

$$log(1 - error(M_i))/(error(M_i)$$

where *error*(M_i) is the error rate of the i-th model. The class with the maximum score after each model votes and weighted accordingly is the 'winner'.

The following common scheme is used in Adaboost. The initial weight for the training tuples are all equal. Iteratively, it will produce the model (M_i) for the i-th round. Sampling with replacement is done but the selection is based on the weight of the training tuples. So, the misclassified ones from previous rounds have more chances of being selected. The classifiers from later rounds are more apt to handle more difficult tuples.

Because the algorithm focus on misclassified tuples, the resulting models often overfit with noisy data. Occasionally, the boosted model may be less accurate than a single model.

In general, both bagging and boosting improved accuracy over a single model. But, there are some studies that suggest that Adaboost may not work well with microarray data. An improvement is suggested in the paper *Boosting and Microarray Data* [33] to increase its performance.

10.5.3 BagBoosting

More recent research, as shown in the paper [31], demonstrates that when bagging is used as a module in boosting, the resulting classifier consistently improves the predictive performance and the probability estimates of both bagging and boosting on real and simulated gene expression data. The overhead for this quasi-guaranteed improvement is a bigger computing effort.

10.6 Evaluating the Accurracy of a Classifier

The validation methods are especially important for microarray data because of the small sample size. The following techniques are commonly used to measure the accuracy and reliability of the classifier:

10.6.1 Holdout and Random Subsampling

The concept of holdout is simple – randomly partition the data into two independent sets, a training set and a test set. Commonly, 2/3 of the data are allocated to training and 1/3 for testing. This automatically reduces the sample size used for training and small sample size is always difficult to train unbiased classifier.

A revised variation is to repeat the process k times and the average of the accuracies obtained with each iteration will be used.

10.6.2 Cross validation/Leave One Out Validation

In k-fold cross-validation, the initial data set is randomly partitioned into k mutually exclusive subsets (or folds). Each of which are of approximately equal size. Then, we iterate k times using each fold as the hold out for validation. So, each fold is used equal number of times for training and one time for testing. For classification, the accuracy is the correct number of classification for k iterations divided by the total number of tuples.

Leave-one-out-cross-validation (LOOCV) is a special case of k-fold with k set to the number of initial tuples. So, only 1 sample is left for testing each time. The study by Golub et al. [6] on leukemia and Shipp et al. on DLBCL [7] use LOOCV to validate their models. Ten-fold cross validation is commonly used in many studies as well.

10.6.3 Bootstrap

Instead of partitioning the learning tuples, the bootstrap methodology samples the initial data set with replacement. So, duplicate tuples can appear in the training set. Tuples that are not selected will form the testing set.

Iteratively, we generate *n* different models by resampling with replacement the initial data set *n* times for the training data set. Testing is done on the samples that are not picked for training. The accuracy of the overall model can be shown to be about [2]

$$Acc(M) = \sum(0.632 \times Acc(M_i)_{test\_set} + 0.368 \times Acc(M_i)_{train\_set})$$

where $Acc(M_i)_{test\_set}$ is the accuracy of the model obtained with bootstrap sample i when it is applied to test set i and $Acc(M_i)_{train\_set}$ is the accuracy of the model obtained with bootstrap sample i when it is applied to the training set i. The bootstrap process works well even with small data sets.

It should be noted that cross-validation after selection of differentially expressed genes from the full data set could result in a highly biased estimate of prediction accuracy. Thus, it is very important to cross-validate all steps of predictor construction in estimating the error rate. Furthermore, additional studies are needed before a classifier can be used in clinical setting [34].

10.7 Comparing Accuracies of Classification Techniques

There are various aspects of the classification of techniques that can be examined. One of the most important performance measurements is accuracy. This section looks at the accuracy of the five commonly used classification techniques: decision trees, k-NN, support vector machines, discriminant analysis and artificial neural network using the corresponding packages available in R. Since there is no R package for the naive Bayesian classifier, it is not included in this comparison. No special tuning and ensemble techniques are applied to increase the prediction accuracies for any particular methods. The main purpose here is not to advocate a particular method but rather to show that accuracies vary significantly based on the parameters chosen. It is the responsibility of the investigator to pick and compare the appropriate methods and the corresponding parameters for a particular study. Other metrics such as Receiver Operating Characteristics (ROC) curves can also be used to evaluate the methods.

10.7.1 Data Set

To compare the efficacy of classification methods, we apply them to two separate publicly available data sets:

1. Diffuse Large B-Cell Lymphoma Outcome Prediction – Expression data of a total of 6817 genes from 58 patients with DLBCL (32 cured and 26 fatal/refractory) are used to predict the outcome of clinical treatment. http://www.broad.mit.edu/publications/broad987s
2. Classification of Bipolar Disorder – Postmortem samples of 61 patients (30 disease and 31 healthy controls) are used to predict bipolar disorder. http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE5388

10.7.2 Experimental Design

We adopt the neighborhood analysis based on the noise to signal ratio

$$abs(\mu_{cured} - \mu_{fatal})/(\sigma_{cured} + \sigma_{fatal})$$

as the criterion for feature selection where μ and σ denote the mean and the standard deviation of the respective class label *fatal*, *cured* for the Shipp dataset. For the biopolar dataset, the class label will be *disease* and *healthy*. The top *n* genes with the highest noise to signal ratio are picked as input to the classification engine. It is the same technique used in both Golub [6] and Shipp [7] studies. For validation, we use the LOOCV scheme to assess the accuracy. The following is the measure being used in the table below:

$$accuracy = \frac{correct\ positive\ prediction + correct\ negative\ prediction}{total\ number\ of\ cases}$$

positive prediction – prediction for “cured” among real “cured” cases

negative prediction – prediction for “fatal/refractory” among real “fatal/refractory” cases.

The LOOCV method is used for the two datasets – one sample is left out and the rest is used as training. The left out sample is used to validate the resulting machine. The process is repeated for all remaining samples. The accuracy as defined above is calculated, plotted in the line graphs (Figs. 10.4 and 10.5) and tabulated in the tables (Tables 10.1 and 10.2) below.

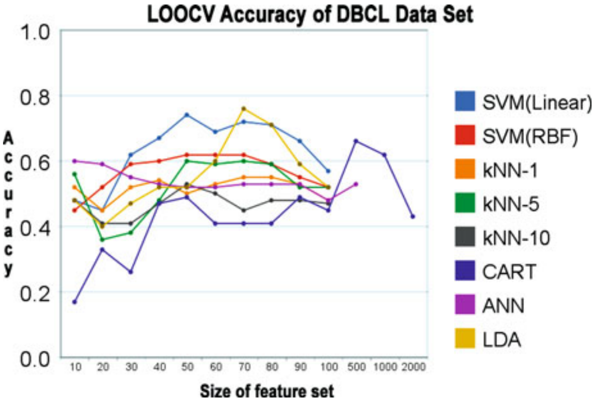


Fig. 10.4 The line graph for LOOCV accuracy of Shipp’s dataset by various classification methods

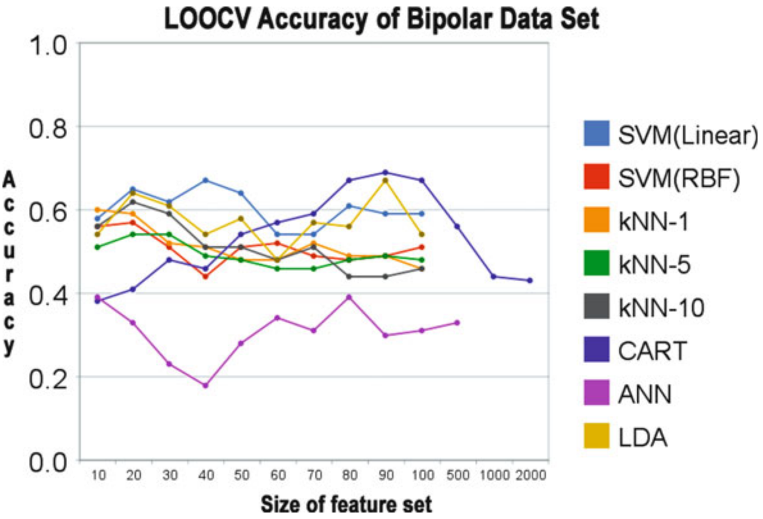


Fig. 10.5 The line graph for LOOCV accuracy of the bipolar dataset by various classification methods

Table 10.1 Overall accuracy of DLBCL outcome prediction

	SVM		k-NN			CART	ANN	DA
	SVM(Linear)	SVM(RBF)	kNN-1	kNN-5	kNN-10	CART	ANN	LDA
10	0.48	0.45	0.52	0.56	0.48	0.17	0.60	0.48
20	0.45	0.52	0.45	0.36	0.41	0.33	0.59	0.40
30	0.62	0.59	0.52	0.38	0.41	0.26	0.55	0.47
40	0.67	0.60	0.54	0.48	0.47	0.47	0.53	0.52
50	0.74	0.62	0.50	0.60	0.53	0.49	0.52	0.52
60	0.69	0.62	0.53	0.59	0.50	0.41	0.52	0.60
70	0.72	0.62	0.55	0.60	0.45	0.41	0.53	0.76
80	0.71	0.59	0.55	0.59	0.48	0.41	0.53	0.71
90	0.66	0.55	0.53	0.52	0.48	0.49	0.53	0.59
100	0.57	0.52	0.48	0.52	0.47	0.45	0.48	0.52
500						0.66	0.53	
1,000						0.62		
2,000						0.43		

Table 10.2 Overall accuracy of bipolar disorder outcome prediction

	SVM		k-NN			CART	ANN	DA
	SVM(Linear)	SVM(RBF)	kNN-1	kNN-5	kNN-10	CART	ANN	LDA
10	0.58	0.56	0.60	0.51	0.56	0.38	0.39	0.54
20	0.65	0.57	0.59	0.54	0.62	0.41	0.33	0.64
30	0.62	0.51	0.52	0.54	0.59	0.48	0.23	0.61
40	0.67	0.44	0.51	0.49	0.51	0.46	0.18	0.54
50	0.64	0.51	0.48	0.48	0.51	0.54	0.28	0.58
60	0.54	0.52	0.48	0.46	0.48	0.57	0.34	0.48
70	0.54	0.49	0.52	0.46	0.51	0.59	0.31	0.57
80	0.61	0.48	0.49	0.49	0.44	0.69	0.30	0.67
90	0.59	0.49	0.49	0.49	0.44	0.69	0.30	0.67
100	0.59	0.51	0.46	0.48	0.46	0.67	0.31	0.54
500						0.56	0.33	
1,000						0.6		
2,000						0.43		

10.7.3 Validation Results

10.7.3.1 DLBCL Study

The highest accuracy is from LDA with 70 genes as predictors, and its accuracy can get up to 0.76. It is closely followed by SVM with linear kernel. With 50 genes as predictors, the accuracy goes up to 0.74. K-NN can only achieve prediction accuracy of 0.60 with 40 genes as predictors. The result differs slightly from the original paper. That could be due to how computation are being done. For example, the original paper uses proprietary software that uses a gradient descent method using noise to signal as criteria to select features that tie in with the classification using SVM. Our computation just use the default method provided by the R package.

10.7.3.2 Bipolar Disorder Study

The LDA and SVM are methods that can achieve above 60% accuracy in some cases. However, none of the classification method do a very good job at prediction for this particular dataset.

10.7.4 Observations

1. The accuracy varies over a large range as the parameters change. In general, it is difficult to obtain the maximum accuracy just by trial and error. Some sort of automatic optimization technique such as gradient descent is needed to obtain the optimal accuracy. The gradients of the function at the current point are estimated and the algorithm moves along the gradient with the steepest descent in order to find a local minimum.
2. There seems to be a delicate balance between feature selection and classification. Different feature selection algorithm will lead to different accuracies.
3. For prediction accuracy, LDA and SVM seems to perform consistently better than the other techniques. However, if ensemble and tuning techniques are applied, the result may be quite different. So, it is likely that investigators choose certain classifier based on their knowledge of the subject matter and familiarity on how to tune the classifier to produce good results.

10.8 Summary of Microarray Classification Software Packages

10.8.1 BRB-ArrayTools

http://linus.nci.nih.gov/BRB-ArrayTools.html

BRB-ArrayTools [35, 36] is a comprehensive Microsoft Excel based GUI package for visualization and statistical analysis of microarray gene expression data. Designed and maintained by NIH, it encompasses several modules for microarray class prediction as well as many other utilities, like survival analysis, time course analysis. It offers class prediction by either single or multiple methods such as compound covariate predictor, Bayesian compound covariate, Diagonal LDA, KNN, SVM and PAM modules. Furthermore, class performance can be optimized by changing several parameters, i.e. gene list, leave-one-out-cross-validation, recursive feature elimination and 0.632+ bootstrap options. ArrayTools is a freeware for academic use with active user support and development team which provides easy-to-use Excel based functionality with robust features to perform microarray raw data processing under one platform. BRB-ArrayTools does not provide command line package and it depends on the R-Excel and R-COM server software. For proper design and analysis of DNA microaray investigations, one can look at the book by Simon et al. [37,38].

10.8.2 Bioconductor

www.bioconductor.org

Packages: MLInterfaces, CMA, MCRestimate

Bioconductor [39, 40] is a R-dependant, open-source collection of software packages for high throughput genome analysis. Designed primarily for professional bioinformaticians, its command line ability together with active open-source development gives great flexibility in data analysis. Depending on the type of package used, it provides many classification and validation methods such as Leave-One-Out-Cross-Validation, K-NN, Monte-Carlo cross-validation, Bootstrap, SVM, Neural Networks, LDA. The CMA package has 21 methods to choose from for classification and has an option to automatically adapt methods to user-submitted data format. Class performance can be fine-tuned using flexible coding for gene filtering, clustering and classification methods. However, with multiple methods to analyze, caution should be ensured to validate results on independent data set rather selecting the best performing method.

10.8.3 GenePattern

http://www.broadinstitute.org/cancer/software/genepattern/desc/expression.html#pred

Designed and maintained at Broad Institute, Gene Pattern [41–43] is a popular free software package for microarray data analysis. Its class prediction module provides GUI for widely used classification methods, i.e. classification and regression trees (CART), K-nearest neighbors (K-NN), Probabilistic Neural Network (PNN), Weighted Voting, and Support Vector Machines (SVM). Unlike Bioconductor, Gene Pattern does not require proficiency in coding. However, many built-in modules work on public server which requires user to upload data in a specified format over internet which could be troublesome for large dataset. Optionally, modules can be installed on local servers.

10.8.4 PAM – Prediction Analysis for Microarrays

http://www-stat.stanford.edu/~tibs/PAM/

PAM is another popular free package for sample classification using nearest shrunken centroid method [44] and added cross-validation support. It has an Excel based plugin and a R package with very easy to use interface, especially automatic gene selection. Even though sample classification is based on only one method, its class prediction performance is nearly matching to that using SVM and it has the lowest average error rate.

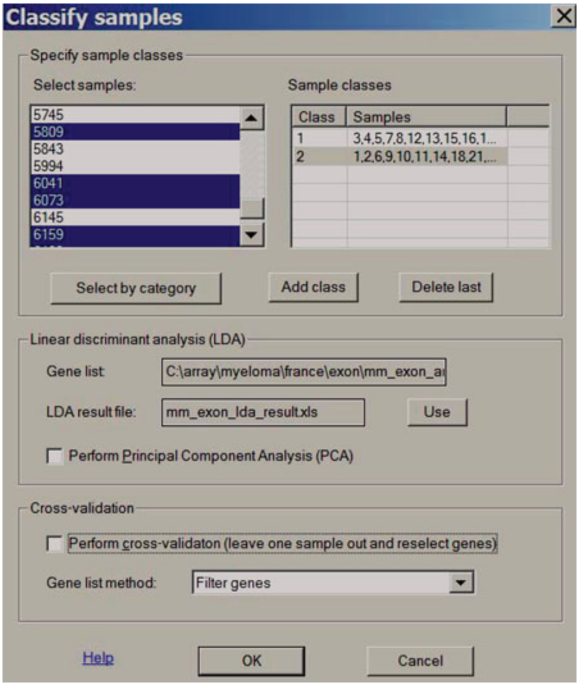
10.8.5 dChip

www.dchip.org

http://www.dchip.org/lda.htm

DNA-Chip Analyzer (dChip) is a Windows software package for probe-level (e.g. Affymetrix platform) and high-level analysis of gene expression microarrays and SNP microarrays [45,46]. dChip uses Linear discriminant analysis (LDA) analysis in R for class prediction. Being a GUI software, dChip provides relatively easy and speedier way of sample classification with further improvement in class prediction performance using Leave-One-Out-Cross-Validation method and ability to filter gene signatures using gene filtering and ANOVA functions. Class prediction function requires R installation and command line support is not included at present.

The dChip class prediction dialog window (Fig. 10.6) can be viewed by selecting Analysis\Classify Samples from menu bar. It provides easy GUI to define classes (on the right side) from available samples in the left side of window. Samples whose



Class	Samples
1	3,4,5,7,8,12,13,15,16,1...
2	1,2,6,9,10,11,14,18,21,...

Gene list: C:\array\myeloma\france\exon\mm_exon_a

LDA result file: mm_exon_lda_result.xls Use

□ Perform Principal Component Analysis (PCA)

□ Perform cross-validaton (leave one sample out and reselect genes)

Gene list method: Filter genes

Fig. 10.6 dChip Class Prediction Dialog Box class is known can be added by clicking “Select by category” option and specifying particular category, i.e. treatment responders, non-responders, etc. Samples not added in any of the class will be regarded as “unknown” samples and their class labels will be predicted after LDA is performed. LDA is performed by providing user-defined gene list obtained from either Analysis\Compare Samples or Analysis\Filter genes functions. Analysis\Compare Samples method gives a list of differential expressed genes between known sample classes and therefore it is preferable method to get higher prediction power in test dataset. To validate the classification performance, check “Perform cross-validation” option and specify the gene list method using filtering or ANOVA function of dChip. Final results of LDA will be stored in an lda result file; an icon will appear in left side panel of dChip, and clicking it will show an LDA scatter plot (see dChip online manual).

10.9 Discussion and Conclusions

The bioinformatics revolution has generated such optimism that in early 2000s, several popular magazines, Wired [47] and The Scientist, [48] reported microarray as the future prognostic tool for many of the life-threatening diseases. Wired even pronounced the end of cancer as we know it. Unfortunately, the good news was premature. A more realistic assessment was discussed in the Fall 2006 Issue of Biomedical Computation Review [49].

Since the groundbreaking research 10 years ago by Golub et al. [6], there have been many additional studies applying to many other different data sets using many different kinds of machine learning techniques for classification. The results are mixed. The accuracy rate depends on the data set. More importantly, some of the results are not reproducible. Tibshirani thought that “a good proportion of the microarray analyses was wrong” [49]. As mentioned earlier, since we only have a limited number of samples but large number of genes, most methods tend to show a relatively high number of false positives as well. So, clinically, we have not been able to apply any of the models we have selected. Further research efforts are needed.

In general, the following questions still need to be answered:

1. Which is the best machine learning classification technique? It is clear from the literature that no one specific technique stands out as the winner. We don’t understand under what conditions should we apply which technique.
2. Are sophisticated classifiers better than simple ones? Some research shows that simple and sophisticated classifiers seem to produce similar results. Even though the sophisticated classifiers produce better results consistently, the simple ones are not far behind. Do all the fine tuning and additional computational needs worth the few percentages of accuracy?
3. What combination of techniques work best? Since we are dealing with a large number of genes, there are multiple selection process in the steps to arrive at a model with a manageable number of genes. It seems that different supervising

and unsupervising techniques can be applied to produce results that are highly accurate. We need to understand various combination of choices when we do the analysis.

4. How to choose some of the parameters? The accuracies achieved by some of the sophisticated techniques require us to adjust certain parameters. For example, we have to pick the appropriate kernel function for SVM; we have to pick k for k-NN; and so on. How do we know we pick the best one or even the right one ? We need a better approach to parameter selection than just trial and error. Dividing data to training, validation, and test sets enables us to select parameters using training and validation data, and apply the best classifier to the test data for an unbiased accuracy estimate.
5. What is the sample size needed to train a classifier? Another parameter that we need to quantify is the sample size. Some research has been done in this area [50]. Since we only have limited number of samples in these experiments, we need to quantify clearly the power of these tests so that we can establish a confidence level when they are in use.
6. Can we apply the research result for clinical usage? Do we need a new approach to the problem? Right now, we cannot apply directly what we have right now for practical clinical usage yet. It is not clear how we are going to get there either. The papers published recently continue to refine machine learning techniques to achieve higher accuracies. But, it is not clear that we have convinced ourselves that it is ready for clinical use. So far we are looking at the details at the gene expression level trying to piece thing together. Do we need a new approach? Are we sure that we are looking at the right objects and abstractions?
7. Research results are not completely reproducible. One of the tenets of scientific progress is the ability to accurately reproduce results. We need to establish proper protocol so that results are reproducible so that we trust the results. Dudoit and Fridlyand concluded that "the tumor classification error rates reported in the literature are generally biased downward, i.e., overestimate the accuracy with which biological and clinical outcomes can be predicted based on expression measures" [2]. Even more alarming is that the analysis may not be done correctly [34]. How can we trust the outcome when we apply this to the prediction of life-threatening diseases? Stricter research protocols are needed so that we can establish credibility from our analysis.
8. In this chapter, the overall accuracy is used as the metric to evaluate the efficacy of the methods. However, other metrics are as important. Two other commonly used metrics are false positive rate and false negative rate. Consider a real patient classified as normal (false negative), missing the treatment; or a normal person classified as patient (false positive), causing worries and unnecessary treatment. The investigators must weigh these metrics according to their needs when decide which methods to use.

Genomic data from microarray and other sources contain much information about cells. Data mining techniques will certainly help to glean valuable information from the data. However, we have yet to arrive at a methodology to consistently classify and predict life-threatening diseases. The information is buried there – we just need to find them. Much progress has been made for the last decade and much is still needed. New technologies such as microRNA profiling and high-throughput sequencing also pose new challenges and opportunities to the genomics-based classification research.

References

1. The Human Genome Project (2003, last modified 2008). *The human genome project home page*. Retrieved from http://www.ornl.gov/sci/techresources/Human_Genome/home.shtml.
2. Speed, T. (Ed.). (2003). *Statistical analysis of gene expression microarray data* (Chap. 3). New York: Chapman & Hall/CRC.
3. NCBI. Dna_microarray (2007). Retrieved from http://www.ncbi.nlm.nih.gov/About/primer/microarrays.html.
4. Piatetsky-Shapiro, G., & Tamayo, P. (Dec 2003). Microarray data mining: Facing the challenges. *SIGKDD Explorations*, *5*(2), 1–5.
5. Chng, W. J., et al. (Apr 2007). Molecular dissection of hyperdiploid multiple myeloma by gene expression profiling. *Cancer Research*, *67*(7), 2982–2989.
6. Golub, T. R., et al. (Oct 15 1999). Molecular classification of cnacer: class discovery and class prediction by gene expression monitoring. *Science*, *286*(5439), 531–537.
7. Shipp, M. A., et al. (Jan 2002). Diffuse large b-cell lymphoma outcome prediction by gene expression profiling and supervised machine learning. *Nature Medicine*, *8*(1), 68–74.
8. Kamber, M., & Han, J. (2006). *Data mining: Concepts and techniques* (2nd ed.). Amsterdam: Elsevier.
9. Moore, A. (2006). Lecture notes on data mining. Retrieved from http://www.autonlab.org/tutorials/.
10. Breiman, L., et al. (1984). *Classification and regression trees*. Belmont, CA: Wadsworth Press.
11. Zhang, H., et al. (2003). Cell and tumor classification using gene expression data: Construction of forests. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(7), 4168–4172, APR.
12. Tan, P. J., Dowe, D. L., & Dix, T. I. (2007). Building classification models from microarray data with tree-based classification algorithms. *AI:2007: Advance in Artificial Intelligence*, 4830.
13. Li, X., & Eick, C. F. (2003). Fast decision tree learning techniques for microarray data collections. *The 2003 International Conference on Machine Learning and Applications*, 2.
14. Peterson, L. E., & Coleman, M. A. (Jan 2008). Machine learning-based receiver operating characteristic (roc) curves for crisp and fuzzy classification of dna microarrays in cancer research. *International Journal of Approximate Reasoning*, *47*, 17–36.
15. Pique-Regi, R., et al. (2005). Sequential diagonal linear discriminant analysis (seqdlda) for microarray classification and gene identification. *Proceedings of the 2005 IEEE Computational Systems Bioinformatics Conf Workshop*.
16. Guo, Y. (2007). Regularized linear discriminant analysis and its application to microarray. *Biostatistics*, *8*(1), 86–100.
17. Vapnik, V. (1998). *Statistical learning theory* (1st ed.). John Wiley and Sons, Inc., Hoboken, New Jersey.
18. Brown, M. et al. (Jan 2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America*, *97*(1), 262–267.
19. Guyon, B., Weston, S., Barnhill, V., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*(1–3), 389–422.
20. Zhang, X., et al. (April 2006). Recursive svm feature selection and sample classification for mass-spectrometry and microarray data. *BMC Bioinformatics*, *7*, 197.

21. Zhang, X., et al. (2006). Gene selection using support vector machines with non-convex penalty. *Bioinformatics 2006*, *22*(1), 88–95.
22. Zhou, X., & Tuck, D. P. (2007). Msvm-rfe: Extensions of svm-rfe for multiclass gene selection on dna microaarray. *Bioinformatics*, *23*(15), 2029.
23. Khan, J. et al. (Jul 2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, *7*, 673–679.
24. O'Neill, M., & Song, L. (2003). Neural network analysis of lymphoma microarray data: prognosis and diagnosis near-perfect. *BMC Bioinformatics*, *4*, 13.
25. Cho, H. S., et al. (2003). cdna microarray data based classification of cancers using neural networks and genetic algorithms. *Nanotech*, *1*, 28–31.
26. Friedman, N., et al. (2000). Using bayesian networks to analyze expression data. *Journal of Computational Biology*, *7*, 601–620.
27. de Ferrari, L., & Aikens, S. (2006). Mining housekeeping genes with a naive bayes classifier. *BMC Genomics*, *7*, 277.
28. Helman, P., et al. (2004). A bayesian network classification methodology for gene expression data. *Journal of Computational Biology*, *11*(4), 581–615.
29. Demichelis, F., et al. (2006). A hierarchical nave bayes model for handling sample heterogeneity in classification problems: An application to tissue microarrays. *BMC Bioinformatics*, *7*, 514.
30. Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140.
31. Dettling, M. (2004). Bagboosting for tumor classification with gene expression data. *Bioinformatics*, *20*(18), 3583–3593.
32. Dudoit, S., & Fridlyand, J. (2003). Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, *19*(9), 1090–1099.
33. Long, P. M., & Bega, V. B. (2003). Boosting and microarray data. *Machine Learning*, *52*(1), 31–44.
34. Simon, R. (2008). Challenges of microarray data and the evaluation of gene expression profile signatures. *Cancer Investigation*, *26*, 327–332.
35. Yanaihara, N., et al. (Mar 2006). Unique microrna molecular profiles in lung cancer diagnosis and prognosis. *Cancer Cell*, *9*(3), 189–198.
36. Bianchi, F., et al. (Nov 2007). Survival prediction of stage i lung adenocarcinomas by expression of 10 genes. *Journal of Clinical Investigation*, *117*(11), 3436–3444.
37. NCI. Review (2003). Retrieved from http://linus.nci.nih.gov/~brb/book.html.
38. Simon, R., et al. (2004). *Design and analysis of DNA microarray investigations*. London-Berlin-Heidelberg: Springer-Verlag.
39. Slawski, M., et al. (Oct 2008). Cma: A comprehensive bioconductor package for supervised classification with high dimensional data. *BMC Bioinformatics*, *9*(1), 439.
40. Golub, T. R., et al. (Oct 1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, *286*(5439), 531–537.
41. Reich, M., et al. (May 2006). Genepattern 2.0. *Nature Genetics*, *38*(5), 500–501.
42. Gadisseur, A., et al. (Jun 2009). Laboratory diagnosis and molecular classification of von willebrand disease. *Acta Haematology*, *121*(2–3), 71–84.
43. Moreno, C. S., et al. (Nov 2005). Novel molecular signaling and classification of human clinically nonfunctional pituitary adenomas identified by gene expression profiling and proteomic analyses. *Cancer Research*, *65*(22), 10214–10222.
44. Tibshirani, R., et al. (Mar 2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences of the United States of America*, *99*, 6567–6572.
45. Li, C., et al. (2001). Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proceedings of the National Academy of Science United States of America*, *98*, 31–36.
46. Lin, M., et al. (2004). dchipsnp: Significance curve and clustering of snp-array-based loss-of-heterozygosity data. *Bioinformatics*, *20*, 1233–1240.
47. Wired. (Aug 2003). The end of cancer (as we know it). *Wired*, *11*, 8.

48. The Scientist. (2004). The making of microarray prognosis. *The Scientist*, *18*(5), 32.
49. Cobb, K. (Fall 2006). Microarrays: The search for meaning in a vast sea of data. *Biomedical Computation Review*, *2*, 17–23.
50. Dobbin, K., & Simon, R. (2005). Sample size determination in microarray experiments for class comparison and prognostic classification. *Biostatistics*, *6*(1), 27–38.

Chapter 11
Statistical Analysis of Single Nucleotide Polymorphism Microarrays in Cancer Studies

Pierre Neuvial, Henrik Bengtsson, and Terence P. Speed

Abstract In this chapter, we focus on statistical questions raised by the identification of copy number alterations in tumor samples using genotyping microarrays, also known as Single Nucleotide Polymorphism (SNP) arrays. We define the copy number states formally, and show how they are assessed by SNP arrays. We identify and discuss general and cancer-specific challenges for SNP array data preprocessing, and how they are addressed by existing methods. We review existing statistical methods for the detection of copy number changes along the genome. We describe the influence of two biological parameters – the proportion of normal cells in the sample, and the ploidy of the tumor – on observed data. Finally, we discuss existing approaches for the detection and calling of copy number aberrations in the particular context of cancer studies, and identify statistical challenges that remain to be addressed.

P. Neuvial
Department of Statistics, University of California, Berkeley, USA
and
Laboratoire Statistique et Génome, Université d'Évry Val d'Essonne, UMR CNRS 8071-USC INRA, France
e-mail: pierre@stat.berkeley.edu

H. Bengtsson
Department of Statistics, University of California, Berkeley, USA
and
Department of Epidemiology & Biostatistics, University of California, San Francisco, USA
e-mail: hb@stat.berkeley.edu

T.P. Speed (✉)
Department of Statistics, University of California, Berkeley, USA
and
Bioinformatics Division, Walter & Eliza Hall Institute of Medical Research, Parkville, Australia
e-mail: terry@stat.berkeley.edu

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_11,
© Springer-Verlag Berlin Heidelberg 2011

11.1 From Biological Questions to Statistical Challenges

Each normal human cell has 23 pairs of chromosomes. For each of them, one chromosome has been inherited from each biological parent. Tumor cells harbor numerous structural alterations of their DNA including point mutations, translocations, small insertion or deletion events, larger scale copy number changes, amplifications, and loss of heterozygosity (LOH), which corresponds to the loss of the contribution of one parent in a genomic region. These alterations can affect genes and regulatory transcripts, which may result in cellular modifications including angiogenesis, immune evasion, metastasis, and altered cell growth, death and metabolism [1]. They are thought to be associated with diagnostic and prognostic factors [2].

An immediate goal of copy number studies in cancer research is to estimate the underlying *copy number state* (to be defined more formally in the next section) at each position along the genome of a tumor sample. Microarray-based technologies have been used for more than a decade to quantify copy numbers at a large number of genomic loci [2–4]. In particular, genotyping microarrays (SNP arrays) are a technology of choice because they combine a high density of markers along the genome (in the order of millions for the current generation) with the ability to assess both changes in total copy number and loss of heterozygosity in a single assay. This is what make them particularly relevant to cancer studies, where both pieces of information are needed to understand the underlying copy number state of the tumor.

In this chapter, we review statistical challenges raised by the analysis of SNP array data in cancer studies. We focus on the analysis of *one* tumor sample. Identifying copy number states from a tumor sample requires *detecting* changes in copy number signals, and *calling* regions, that is, assigning a copy number state to each region detected. The main ingredient for the detection part is the fact that DNA copy number is *locally constant* along the genome: *locus-level* estimates can thus be combined in *region-level* estimates. However, for this property of local constancy to be fully exploited, SNP array data first have to be pre-processed so that locus-level estimates for a given sample are comparable across loci. For the calling step to be performed satisfactorily, biological factors that influence the estimated copy number levels – tumor ploidy and normal contamination – have to be understood and acknowledged for.

11.1.1 Outline

We begin by defining the copy number states of interest in cancer studies, and showing how estimates can be obtained from preprocessed SNP array data for each locus (Sect. 11.2). We then describe current methods for SNP array data preprocessing, with a focus on specific challenges for copy number studies in cancers (Sect. 11.3). In Sect. 11.4 we review statistical methods that have been proposed to combine locus-level copy number estimates (as obtained after preprocessing) to detect copy number changes along the genome. In Sect. 11.5 we describe the influence of tumor ploidy and normal contamination on observed signals and their interpretation. In Sect. 11.6 we show how the methods described in Sect. 11.4 have been applied to SNP array data in cancer studies, by accounting for or taking advantage of the characteristics of the data described in Sect. 11.5. We conclude by identifying ongoing challenges for the statistical analysis of SNP array data in cancer studies (Sect. 11.7).

11.2 Minor and Major Copy Numbers in Cancer Studies

We define the *copy number state* of a tumor at a given genomic locus j as a pair of numbers $(\underline{\gamma}_j, \overline{\gamma}_j)$, where $\underline{\gamma}_j \geq 0$ and $\overline{\gamma}_j \geq 0$ are respectively the smaller and the larger of the two parental copy numbers at this locus. By definition we have $\underline{\gamma}_j \leq \overline{\gamma}_j$, and $\gamma_j = \underline{\gamma}_j + \overline{\gamma}_j$ is the total copy number. The quantities $\underline{\gamma}_j$ and $\overline{\gamma}_j$ are called minor and major copy numbers, respectively. Note that $\underline{\gamma}_j$, $\overline{\gamma}_j$, and γ_j need not be whole numbers, especially because of the possible presence of normal cells in the tumor sample. This point is explained in detail in Sect. 11.5.

The two-dimensional vector $(\underline{\gamma}_j, \overline{\gamma}_j)$ does not characterize parental copy numbers at locus j in the tumor. Indeed, the information of which of minor or major copy numbers corresponds to the maternal chromosome at locus j, and which one corresponds to the paternal chromosome is missing from $(\underline{\gamma}_j, \overline{\gamma}_j)$, and it may change across loci. In short, because of the constraint $\underline{\gamma}_j \leq \overline{\gamma}_j$, minor and major copy numbers (CNs) are not *phased* in terms of parental copy numbers.

The remainder of this section is organized as follows. In Sect. 11.2.1 we focus on *true* copy number signals, that is, the actual copy numbers in the biological samples. We demonstrate that knowing true minor and major copy numbers is enough to characterize copy number events of interest in cancer studies. In Sect. 11.2.2 we show that true copy numbers, including minor and major copy numbers, can be *estimated* from SNP array data at the locus level. Notation used in the chapter is summarized in Sect. 11.2.3.

11.2.1 Information Relevant to Copy Number Studies in Cancers

Table 11.1 summarizes the copy number states relevant to cancer studies in terms of minor and major copy numbers. They are described as the conjunction of information regarding total copy numbers and (loss of) heterozygosity. For example, knowing the total copy number in a region of LOH ($\underline{\gamma} = 0$) allows us to distinguish between hemizygous deletions $(\underline{\gamma}, \overline{\gamma}) = (0, 1)$, that is, single copy deletions, from LOH when the total copy number is two $(0, 2)$, so-called copy-neutral LOH or acquired uniparental disomy. Conversely, among regions of neutral copy number ($\gamma = 2$), regions of copy-neutral LOH $(0, 2)$ can be distinguished from normal regions $(1, 1)$ based on the LOH status of the region. This is distinction is important for data interpretation, as copy-neutral LOH is a known mechanism through which

Table 11.1 Minor and major copy number states of interest for cancer studies, presented as the conjunction of information regarding total copy number (*columns*) and heterozygosity status (*rows*)

	Deletion	Neutral	Gain
Loss of heterozygosity	$(0, 1)$	$(0, 2)$	$(0, \overline{\gamma})$ with $\overline{\gamma} \geq 3$
Heterozygosity	$(0, 0)$	$(1, 1)$	$(\underline{\gamma}, \overline{\gamma})$ with $1 \leq \underline{\gamma} \leq \overline{\gamma}$ and $\underline{\gamma} + \overline{\gamma} > 2$

a recessive tumor suppressor gene can be expressed with no apparent change in total copy number [5].

Regions of LOH are characterized by the absence of one of the two parental chromosomes, that is, by a null minor copy number: $\underline{\gamma}_j = 0$. However, (loss of) heterozygosity is a binary concept which can be insufficient (even when combined with total copy numbers) to fully characterize subtle copy number events such as complex gains, as in the lower right cell of Table 11.1 which corresponds to a copy number gain with retention of heterozygosity. For example, $(1, 3)$ and $(2, 2)$ are two states that fall into this category, with the same total copy number. However, the biological interpretation of these two states can be quite different: $(2, 2)$ is a balanced duplication of a chromosomal region, while $(1, 3)$ corresponds to an allele-specific amplification, which can typically pinpoint regions containing oncogenes.

This example illustrates the need for a quantitative measure to characterize *allelic imbalance* between parental copy numbers at a given locus, rather than a binary variable (retention or loss of heterozygosity). Several closely related measures have been proposed to quantify allelic imbalance in cancers [6–8]. These measures can be written in terms of minor and major copy numbers and quantify the distance to the heterozygous status. In this chapter, we denote the *allelic imbalance* at locus j by $\delta_j \in [0, 1]$, and use the following definition:

$$\delta_j = \frac{\overline{\gamma}_j - \underline{\gamma}_j}{\overline{\gamma}_j + \underline{\gamma}_j}. \tag{11.1}$$

In the above example, $(\underline{\gamma}, \overline{\gamma}) = (2, 2)$ yields $\delta = 0$ (allelic balance or heterozygosity), while $(1, 3)$ yields $\delta = 1/2$ (*partial* loss of heterozygosity). Note how a hemizygous deletion $(0, 1)$ and a copy-neutral LOH $(0, 2)$ both yield $\delta = 1$.

11.2.2 What can be Estimated from SNP Array Data

Single Nucleotide Polymorphisms (SNPs) are genomic positions where the DNA sequence varies at a substantial rate across individuals of some population. For most SNPs only two (out of four) variants are observed. These variants are called *alleles* and arbitrarily denoted by A and B. SNP arrays are a microarray-based technology which targets both alleles of a large number of SNPs. Although they were originally developed for genotyping studies, they have also been proved quite useful for copy number studies, especially in cancers.

Current generations of SNP arrays (Affymetrix GenomeWideSNP_6 and Illumina Human1M-Duo) interrogate approximately one million SNPs, that is, of the order of 10% of the total number of known human SNPs. They also incorporate copy number probes, which measure total copy numbers at non-polymorphic loci for increased resolution of copy number studies. We refer to [9] for a more comprehensive review on SNP array technologies. Specific characteristics of SNP array assays that are relevant to the data analysis and particularly to data preprocessing are explained in more detail in Sect. 11.3.

For the present section it is sufficient to note that SNP array data (after preprocessing as explained in Sect. 11.3) can be summarized by a two-dimensional vector $(c_j, b_j)_{j \in \mathscr{J}}$ of *locus-level estimates*, where $\mathscr{J}$ denotes the set of J loci targeted by the microarray. When j is a SNP, c_j is the sum of the contribution of the two alleles at j called allele-specific copy numbers, and b_j is the corresponding fraction of signal coming from allele B at j. Following [6, 10, 11], b_j will be called *allele B fraction*. The corresponding allele A fraction is $a_j = 1 - b_j$. The corresponding allele-specific copy numbers A and B can therefore be written as $(a_j c_j, b_j c_j)_{j \in \mathscr{J}}$. When j is a copy number probe, c_j is the total intensity signal at j, while b_j and a_j are not defined.

Figure 11.1 shows Affymetrix GenomeWideSNP_6 data 50 Mb-long genomic region on Chromosome 2 of an ovarian tumor sample from the Cancer Genome Atlas (TCGA). TCGA is a collaborative initiative to provide a high-throughput molecular characterization of a large number of tumors from different cancer types, with the goal to improve biological understanding and clinical treatment of these cancers [12, 13]. These data have been preprocessed using an allele-specific version of the CRMAv2 method [14], called AS-CRMAv2, followed by the TumorBoost method [15] for normalization of raw allele-specific copy numbers.

Previous copy number analyses led by TCGA have shown that this tumor has two copy number transitions in this region. The first one occurs at ~124.2 Mb, between a normal region: $(\underline{\gamma}, \overline{\gamma}) = (1, 1)$ and a region of single chromosome gain: $(\underline{\gamma}, \overline{\gamma}) = (1, 2)$. The second transition occurs at ~140.9 Mb, between a region of single gain and a region of copy-neutral LOH: $(\underline{\gamma}, \overline{\gamma}) = (0, 2)$.

11.2.2.1 Obtaining Locus-Level Estimates of Minor and Major Copy Numbers, and Allelic Imbalances

For any configuration of the paternal and maternal genotypes at SNP j, true allelic ratios α_j and β_j satisfy

$$(\alpha_j, \beta_j) \in \left\{0, \underline{\gamma}_j/\gamma_j, \overline{\gamma}_j/\gamma_j, 1\right\}, \tag{11.2}$$

with the constraint $\alpha_j + \beta_j = 1$. In particular, if SNP j is heterozygous in the germline, then by definition the alleles inherited from the two parents at this locus differ, and the minimum and maximum allelic ratios satisfy $\min(\alpha_j, \beta_j) = \underline{\gamma}_j/\gamma_j$ and $\max(\alpha_j, \beta_j) = \overline{\gamma}_j/\gamma_j$. Therefore, minor and major copy numbers may be estimated as the locus level by

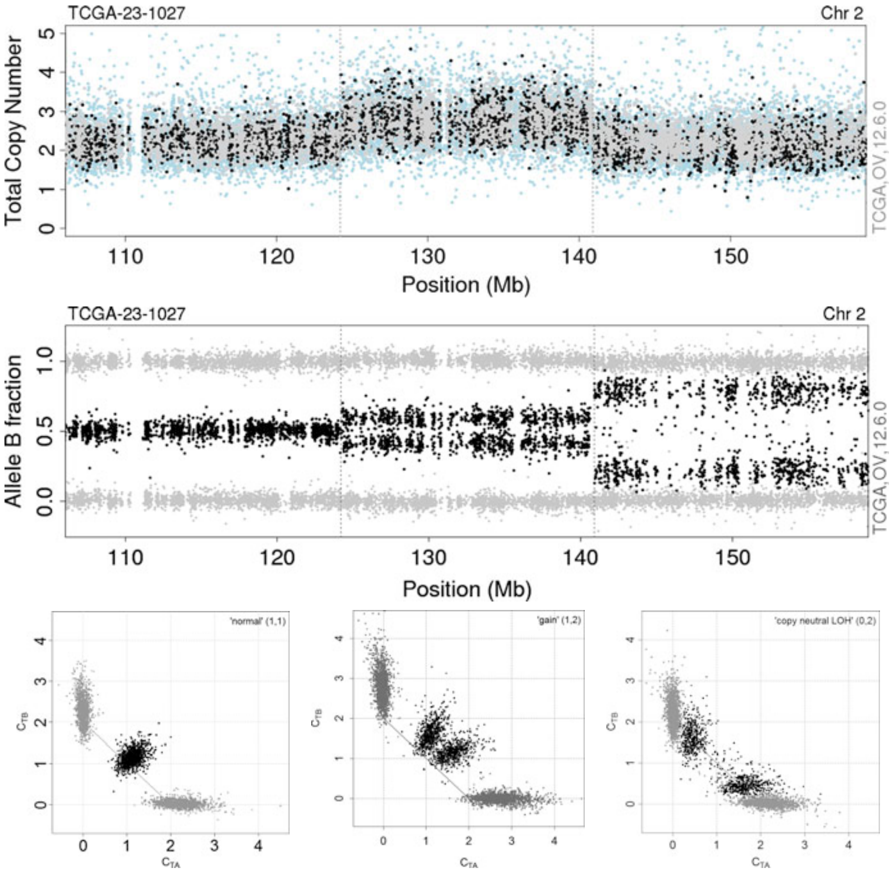


Fig. 11.1 Locus-level estimates from Affymetrix GenomeWideSNP_6 data in three copy number regions on chromosome 2 of a TCGA ovarian tumor sample: normal $(1, 1)$, gain $(1, 2)$ and copy-neutral LOH $(0, 2)$. *Top panel*, total copy numbers (c_j) along chromosome 2. *Middle panel*, allelic ratios (b_j) along chromosome 2. Transitions between the three copy number states are indicated by dashed gray vertical lines in the top and middle panels. *Bottom panels*, allele-specific copy numbers: $(a_j c_j, b_j c_j)$ in each of the three regions. *Gray*: SNPs called homozygous in the paired normal sample, and copy number probes; *black*: SNPs called heterozygous in a paired normal sample (not shown). The data were preprocessed using AS-CRMAv2 [14] followed by TumorBoost [15]

$$\begin{cases} \underline{c}_j & = c_j \cdot \min(a_j, b_j) \\ \overline{c}_j & = c_j \cdot \max(a_j, b_j) \end{cases} . \tag{11.3}$$

The true allelic imbalance as defined in Eq. 11.1 may then be written as $\delta_j = 1 - 2 \cdot \min(\alpha_j, \beta_j)$, and the corresponding locus-level estimate becomes

$$d_j = 1 - 2 \cdot \min(a_j, b_j). \tag{11.4}$$

Table 11.2 Notation: true copy numbers and corresponding locus-level estimates from SNP arrays

	True	Locus-level estimate	Locus type
Total copy number	γ_j	c_j	SNP and CN
Allele A fraction	α_j	a_j	SNP
Allele B fraction	$\beta_j = 1 - \alpha_j$	$b_j = 1 - a_j$	SNP
Minor copy number	$\underline{\gamma}_j = \gamma_j \cdot \min(\alpha_j, \beta_j)$	$\underline{c}_j = c_j \cdot \min(a_j, b_j)$	Heterozygous SNP
Major copy number	$\overline{\gamma}_j = \gamma_j \cdot \max(\alpha_j, \beta_j)$	$\overline{c}_j = c_j \cdot \max(a_j, b_j)$	Heterozygous SNP
Allelic imbalance	$\delta_j = (\overline{\gamma}_j - \underline{\gamma}_j)/\gamma_j$	$d_j = 1 - 2 \cdot \min(a_j, b_j)$	Heterozygous SNP

11.2.3 Notation

The notation used in this chapter for true copy number signals (Greek letters) and the corresponding *locus-level* estimates (Roman letters) is gathered in Table 11.2.

11.3 Preprocessing

The goal of this section is to explain how *locus-level estimates* for total copy numbers (c_j) and allelic ratios (a_j and $b_j = 1 - a_j$), as defined in Sect. 11.2, can be obtained from the observed signal intensities retrieved from SNP array experiments. We focus on the two main SNP array platforms, which are manufactured by Affymetrix [16, 17] and Illumina [18–20]. The first steps that have to be carried out for low-level analysis of microarray data consist in correcting data for sources of unwanted variation, in order to make observed signals *comparable across samples for a given locus*. These steps are described in Sect. 11.3.1. We note that the methods described in this section are generally technology-specific, but not specific to cancer studies – they are relevant to any SNP array data analysis. In cancer studies however, the observed signals also need to be *comparable across loci for a given sample*, so that downstream analysis methods can take advantage of the local constancy of the signal along the genome to combine locus-level estimates into region-level estimates. This question is addressed in Sect. 11.3.2. Note that because the methods developed in Sect. 11.3.2 rely on *reference samples* for the estimation of copy numbers, their application requires making signal intensities comparable across samples (as explained in Sect. 11.3.1) in the first place. Section 11.3.2 is not technology-specific; however it is only relevant to copy number studies in cancers.

11.3.1 Making Signals Comparable Across Samples

SNP arrays were originally developed and used for genotyping purposes in genome-wide association studies (GWAS). Genotype calls are generally estimated independently for each SNP, by comparing the distribution of allelic signals across samples. Necessarily, preprocessing methods for SNP arrays were initially focused on making signals comparable across samples. In this section, we briefly review the design principles of existing methods addressing this point. As Affymetrix and Illumina assays are quite different, these methods are mostly platform-specific.

11.3.1.1 Affymetrix

A variety of preprocessing methods have been suggested for Affymetrix SNP arrays, e.g. (implicit or explicit) background correction, allelic-crosstalk calibration, probe-sequence normalization, PCR fragment-length normalization, several distribution-based normalization methods, and various methods summarizing probe-level signals into locus-level estimates.

Correction of PCR and Sequence Related Effects

Affymetrix genotyping assays involve a Polymerase Chain Reaction (PCR) amplification step [16,17]. In the assays where restriction enzymes are used to fragment the target DNA, the locus-specific copy number estimates may be correlated with the fragment length. Since the fragments are known from the genome annotation it is straightforward to estimate and correct for such effects [14,21–24]. Moreover, it has been reported that observed intensities are also correlated to the GC content [21,24]. More complex relationships with the nucleotide sequences of the probes [14, 23] have been observed as well.

As these parameters may vary across assays and between hybridizations, they need to be corrected for in order to make comparisons across samples meaningful and more precise. Existing approaches typically involve non-linear regression of signal intensities on PCR fragment length, GC content and nucleotide position [14, 21–24].

Generic Probe-Level Normalization

Generic probe-level normalization is a crucial step of microarray preprocessing which aims at making probe signals comparable between samples. For Affymetrix data, methods originally developed for the preprocessing of expression microarray data – lowess normalization [25], invariant-set normalization [26] or quantile normalization [27], have been successfully applied to SNP array data. These approaches explicitly constrain probe-levels signals to be comparable across arrays.

Correction for Allelic Crosstalk

It has been recently shown that most of the non-biological differences between the distribution of probe-level signals across samples could be attributed to *allelic crosstalk* (including an offset correction), that is, cross-hybridization between probes targeting the two alleles of a SNP [24,28]. One advantage of allelic crosstalk calibration is that it effectively makes probe-level signals comparable across samples without imposing constraints on intensity distributions [14, 24]. It can also be applied to each array separately.

Summarization of Probe-Level Signals

Summarization combines normalized probe-level signals into locus-level estimates by fitting a log-additive or multiplicative model of the intensities. These models were first developed for the analysis of oligonucleotide expression microarrays [26,27] and later adapted to SNP arrays [23,29,30]. Related multi-array models that explicitly model allelic crosstalk at the summarization step have also been suggested [28, 31].

A common feature of Affymetrix SNP arrays is that each SNP is associated with a set of 25 nucleotide-long probe sequences. Half of these *probe sets* target allele A and the other half target allele B. However, the technology has evolved substantially across generations of SNP arrays, as a result of an effort from both the manufacturer and the scientific community [9]. With the latest generation of SNP arrays (GenomeWideSNP_5 and 6), all probes targeting a given allele-specific or total copy number locus are technical replicates. With this simplified probe set design, using the median of replicated probes within an array as a summary has been shown [14] to perform as good as or better than previously proposed summarization models that required several arrays to be used.

11.3.1.2 Illumina

Almost all studies performed using Illumina data use the preprocessing method provided by Illumina's BeadStudio software [10, 32], which is an affine transformation of the original data that corrects for offset and signal compression (or allelic crosstalk), and scales the data based on control points. The parameters for this affine transformation are estimated independently for each sample, for each *sub-bead pool*. As the Infinium assay does not involve PCR amplification, correction for sequence effects is not needed for Illumina SNP arrays.

Recent works demonstrated that the signals after BeadStudio normalization suffer from a dye bias [33]: the distribution of normalized signals differ substantially between the two types of fluorescent dyes (Cy3 and Cy5) that are used in the Infinium II assay [34]. The correction method proposed by [33] consists in applying quantile normalization [27] to the normalize the two dyes. Importantly, this is still done independently for each array.

11.3.2 Making Signals Comparable Across Probes

Signal intensities at a given locus j can be assumed to be proportional to the corresponding true copy numbers, but the proportionality coefficient is unfortunately locus specific and unknown [26,27,35]. These coefficients are known as *locus affinities*. In copy number studies, true copy numbers are expected to be locally constant along the genome. This property is exploited by downstream segmentation methods to detect copy number changes along the genome, as explained in Sects. 11.4 and 11.6. It is therefore fundamental for these downstream analyses that these locus affinities be canceled beforehand, in order to make copy number signals comparable across neighboring loci. This section describes how existing methods address this question for total copy number and allelic signals.

11.3.2.1 Total Copy Numbers

As locus affinities are not sample-specific, they can be effectively canceled from total signals by dividing the observed(summarized) signal intensity y_j at locus j by an observed *reference* signal intensity, $y_j^{(R)}$, at the same locus, which is obtained from a sample or a pool of samples for which the true copy number at locus j, $\gamma_j^{(R)}$, is known:

$$c_j = \gamma_j^{(R)} \frac{y_j}{y_j^{(R)}}. \tag{11.5}$$

In general the reference is chosen to be copy-number neutral ("copy neutral"), that is, so that $\gamma_j^{(R)} = 2$ for $j \in \mathscr{J}$. There are several choices of total reference signal $y_j^{(R)}$, depending on the study design [36,37]. For instance, in a paired tumor-normal study, the reference signal at a given locus may be the corresponding total signal from a matched normal tissue sample or normal blood sample, whereas in a tumor study without matched normals, it may be the corresponding robust average (e.g. a median) of all samples in the study. If some of the samples in the study are normal samples, their robust average may be used as a reference instead.

It is in general better to use a reference from the same lab as the test sample, and possibly from the same batch of arrays. This is illustrated by Fig. 11.2, where three different sets of cytogenetically normal samples were used as references for the same tumor SNP array. The tumor SNP array is from a breast cancer cell line hybridized at the Lawrence Berkeley National Laboratory (LBNL). All samples were hybridized on the Affymetrix GenomeWideSNP_6 platform, normalized using CRMAv2 [14]. Copy number profiles were segmented using the Circular Binary Segmentation (CBS) method [38]. The figures were generated using ChromosomeExplorer within the aroma.affymetrix framework [39].

The signals in the three panels of Fig. 11.2 are of similar amplitude: the difference between copy number estimates (black segments) between two successive copy number regions is comparable across panels. Therefore, signal to noise ratios can be compared on the basis of the corresponding noise levels. We quantified the noise level (along the whole genome) for each choice of a reference using a robust first-order standard deviation estimator [40,41]:

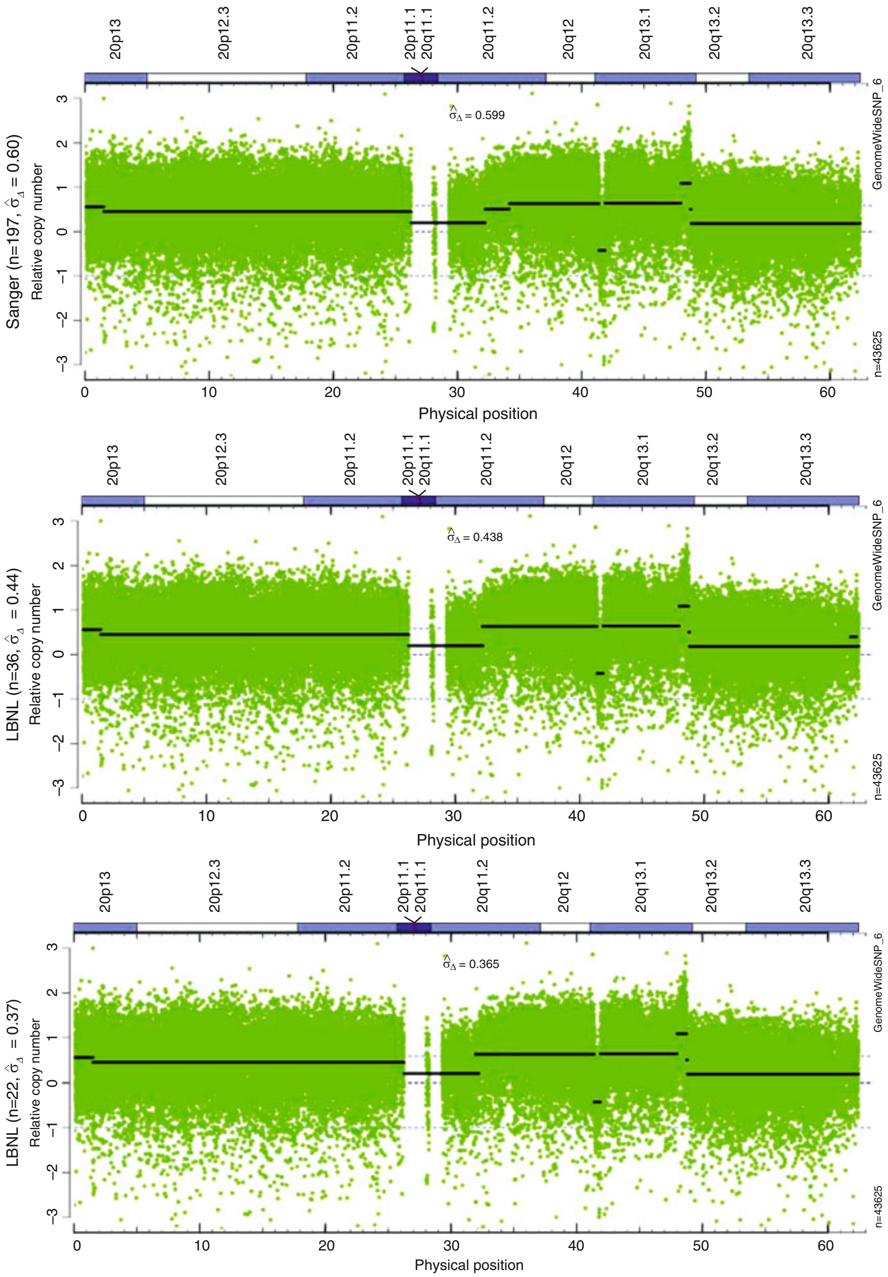


Fig. 11.2 Influence of the choice of a reference sample on the signal to noise ratio in total copy number signals. Three different sets of normal references are used to estimate total copy numbers for the same tumor SNP array hybridized at the Lawrence Berkeley National Laboratory (LBNL): 197 samples from another lab (*top panel*), 36 arrays from LBNL (*middle panel*), and 22 arrays from LBNL, and the same batch as the tumor sample (*bottom panel*). Dots: locus-level estimates; segments: region-level estimates after segmentation by CBS [42]

$$\widehat{\sigma}_{\Delta} = \frac{1}{\sqrt{2}} \cdot \Phi^{-1}(3/4) \cdot \operatorname*{median}_{j}(|z_j - \operatorname*{median}_{j'}(z_{j'})|), \tag{11.6}$$

where $z_j = c_{j+1} - c_j$ for $j = 1, \dots, J-1$. The scaling factors $1/\sqrt{2} \approx 0.7071$ and $\Phi^{-1}(3/4) \approx 1.4826$ make $\widehat{\sigma}_{\Delta}$ a consistent estimator of the c_j under the assumption that c_j, and hence z_j, is Gaussian and i.i.d. Because this estimator relies on the first order differences z_j, it is robust against change points and can therefore be used without knowing where the true change points are.

The noise level is high when samples from a different lab are used as references (top panel: $\widehat{\sigma}_{\Delta} = 0.60$), even when the number of samples is large (197). It is substantially smaller when references from the same lab (LBNL in this particular example) are used (middle panel: $\widehat{\sigma}_{\Delta} = 0.44$), even in a much smaller number (36). It is even lower when references from the same *batch* of arrays (bottom panel: $\widehat{\sigma}_{\Delta} = 0.37$): in this example, the reference set consisted of only 22 arrays hybridized on the same day as the tumor sample.

11.3.2.2 Allelic Ratios

Allelic ratios for a given SNP j are usually estimated as the ratio of the signal intensity of one allele relative to the total signal intensity. For B allele fractions, this yields

$$b_j = \frac{y_{jB}}{y_j}, \tag{11.7}$$

where $y_j = y_{jA} + y_{jB}$, and y_{jA} and y_{jB} are the observed signal intensities for allele A and B, respectively. Note that contrary to total signals, no external reference is needed at this stage: allelic ratios can be estimated from a single hybridization. However, these estimates have been reported to suffer from systematic deviations from their corresponding true values [10, 15, 20, 43]. One possible explanation for this effect is that locus affinities are not only locus-specific but *allele-specific*, so that they may not be adequately canceled by the ratio in Eq. 11.7. Several approaches have been developed to normalize raw allelic ratios based on paired or unpaired normal reference hybridizations, greatly improving the signal to noise ratio for downstream analyses for Illumina data [10, 20], or both Affymetrix and Illumina data [15]. This is illustrated by Fig. 11.3 for the TumorBoost method [15].

11.4 Copy Number Change Detection: From Locus-Level to Region-Level Estimates

Copy number profiles in tumors are consequences of genomic events at the regional scale, such as small or large deletions or gains. Therefore, true copy number signals can safely be modeled as locally constant in tumor samples. This assumption is one

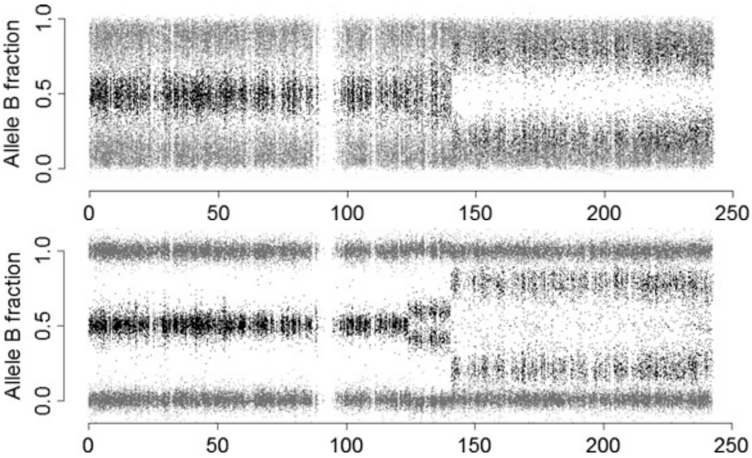


Fig. 11.3 Improved signal to noise ratio after normalization by the TumorBoost method [15]. *Top*: raw allelic ratios as in Eq. 11.7. *Bottom*: TumorBoost-normalized allelic ratios, using allelic ratios from of a paired normal hybridization. Data is taken from the same tumor sample and chromosome as in Fig. 11.1

of the bases of all the algorithms that have been proposed for detecting copy number changes from microarray data. The goal of this section is to explain how locus-level copy number estimates (obtained after preprocessing as described in Sect. 11.3) can be combined to *detect copy number changes* along the genome.

The methods described in this section can be used to segment total, minor and major copy numbers, or allelic imbalances: these applications are discussed in Sect. 11.6. For simplicity of notation and vocabulary, we will loosely refer to *copy numbers* and use the notation c for locus-level estimates, and γ for the corresponding true values.

Two main types of methods have been developed and are used in practice: change-point models and Hidden Markov Models (HMM). In the context of copy number analyses, they were initially applied to microarray technologies that only assess *total* signals, in particular array Comparative Genomic Hybridization (array-CGH) [3]. The practical performance of these methods has been reviewed in [44,45]. The present section provides an up to date statistical review of currently available methods for copy number segmentation using change point or Hidden Markov Models.

For simplicity, we will only use genomic positions $j = 1, 2, \ldots, J$ corresponding to the ordering of loci, rather than the physical location (in basepairs) of the loci, in the following discussion and equations. This is also the most commonly used approach in existing methods. Incorporating physical locations as well introduces another level of complexity to the notation and the models that is unnecessary for the overview presented here.

11.4.1 Change-Point Models

We assume that there exists a partition of the genome into K segments, $k = 1, 2, \dots, K$, such that true copy numbers are constant in each segment. Specifically, there exists an index vector of $K + 1$ loci $\mathbf{t}(K) = (t_k)_{0 \le k \le K}$ called *change points*, such that $1 = t_0 < t_1 < \cdots < t_{K-1} < t_K = J$, and an associated vector of K *region-level true copy numbers* $\boldsymbol{\Gamma} = (\Gamma_k)_{1 \le k \le K}$ such that true copy numbers $\boldsymbol{\gamma} = (\gamma_j)_{j \in \mathscr{J}}$ are constant equal to Γ_k in the interval $[t_{k-1}, t_k)$. That is,

$$\gamma_j = \Gamma_k \,;\; \forall j \in [t_{k-1}, t_k), \forall k \in \{1, \dots, K\}. \tag{11.8}$$

Letting $k(j)$ be the largest index k such that $t_k \le j$, the observation $\mathbf{c} = (c_j)_{j \in \mathscr{J}}$ may then be modeled as

$$c_j = \Gamma_{k(j)} + \varepsilon_j, \tag{11.9}$$

where the errors $(\varepsilon_j)_{j \in \mathscr{J}}$ are independent and identically distributed (i.i.d.), and generally assumed to be Gaussian ($\mathscr{N}(0, \sigma^2)$). When the number K of segments and the vector $\mathbf{t}(K) = (t_0, \dots, t_K)$ of change point locations are known, the log-likelihood $\ell(K, \mathbf{t}(K), \boldsymbol{\gamma}; \mathbf{c})$ of the model described by Eq. 11.9 is additive in each segment:

$$\ell(K, \mathbf{t}(K), \boldsymbol{\gamma}; \mathbf{c}) = J \log(2\pi\sigma^2) + \frac{1}{\sigma^2} \sum_{k=1}^{K} \sum_{j \in [t_{k-1}, t_k)} \left(c_j - \Gamma_{k(j)}\right)^2. \tag{11.10}$$

In this idealized situation, the maximum likelihood estimator of each Γ_k is the empirical mean of the observed signals within the k^{th} segment. In practice though, both K and $\mathbf{t}(K)$ are unknown, which gives rise to a model selection problem (choosing K), and a combinatorial problem: choosing $\mathbf{t}(K)$ for a given K. Indeed, the number of possible configurations for $\mathbf{t}(K)$ is $\binom{K-1}{J-1}$, that is, $O(J^{K-1})$, which is prohibitively large in realistic situations where K is in the dozens and J is currently of the order of 10^5–10^6.

11.4.1.1 Heuristics

The first approach taken to address these issues has been to combine a Bayesian Information Criterion (BIC) penalization for the choice of K with a genetic programming algorithm for the choice of $\mathbf{t}(K)$ [46]. Three main directions have been explored to improve on this early attempt. The most widely used method in practice, known as Circular Binary Segmentation, implements a greedy approach which recursively looks for the best partition of the data into two (or three) segments [38]. The depth of the recursion is determined by the significance of the change points, which implicitly determines K. This step has been made faster using permutation techniques in a second version of the method [42], which has been used to produce the segmentation obtained in Fig. 11.4. A modified BIC criterion has also been proposed to estimate K directly [47].

11.4.1.2 Exact Solutions

Second, several methods have been proposed that solve the original problem exactly. First, one can take advantage of the additivity of the log-likelihood in the segments and use dynamic programming to reduce the complexity of the exhaustive search for the best $\mathbf{t}(K)$ for a given K from $O(J^{K-1})$ to $O(K{\cdot}J^2)$. This idea has been combined with an adaptive penalization method [48] to build a quadratic ($O(K{\cdot}J^2)$) change point detection algorithm [49]. Such a method cannot be used to segment DNA copy number profiles from the latest generations of microarrays, for which more than 10^6 loci can be interrogated. A pruned dynamic programming algorithm has been proposed recently that recovers the optimal solution much faster [50]. Although its worst case complexity is still $O(K{\cdot}J^2)$, in practical situations it is almost linear in J, which makes it quite appealing for current copy number segmentation problems.

11.4.1.3 Convex Relaxations

A third direction uses *convex relaxation*, which is a classical approach in statistical machine learning. It consists in replacing a non-convex optimization problem by a slightly different, but convex, version of the problem, which can be solved efficiently. Two regression methods based on Lasso-type penalties have been applied to the problem of detecting changes in DNA copy number signals [51,52].

The first method [51] is an adaptation of the Fused Lasso [53], which solves the constrained optimization problem

$$\min_{(\gamma_j)_{1\leq j\leq J}} \sum_{j=1}^{J} \left(c_j - \gamma_j\right)^2 \quad \text{s.t.} \quad \sum_{j=1}^{J-1} |\gamma_{j+1} - \gamma_j| \leq v \text{ and } \sum_{j=1}^{J} |\gamma_j - 2| \leq u. \tag{11.11}$$

Formally, this method constrains the ℓ_1 norm of the jumps in $\boldsymbol{\gamma}$, which can be seen as a convex relaxation of constraining the number of jumps (that is, the ℓ_0 norm of the jumps). In words, the mean amplitude of the changes in estimated copy-number levels ($|\gamma_{j+1}-\gamma_j|$) is not allowed to be too large. Moreover, this model incorporates a sparsity constraint on $|\gamma_j - 2|$ enforcing that most loci correspond to the copy-neutral state, where 2 represents the copy number of the copy-neutral state. For non-diploid copy-neutral state, this copy-number level should be adjusted accordingly. The complexity of the algorithm proposed in [51] is (at best) quadratic in the number of data points, that is $O(J^2)$, which is too expensive for recent data sets.

The second method [52] is a relaxed version of Eq. 11.11 where only the amplitudes of the changes are constrained, resulting in the constrained optimization problem

$$\min_{(\gamma_j)_{1\leq j\leq J}} \sum_{j=1}^{J} \left(c_j - \gamma_j\right)^2 \ \text{ s.t. } \sum_{j=1}^{J-1} |\gamma_{j+1} - \gamma_j| \leq \nu. \tag{11.12}$$

This optimization problem can be written as a Lasso-type regression problem and can therefore be solved in $O(K^3 + J \cdot K^2)$ using a Least Angle Regression (LARS) algorithm [54] to select the first K change points. The authors of [52] suggest to prune the obtained set $\mathbf{t}(K)$ of candidate change points by running the aforementioned dynamic programming algorithm on the set of partitions consisting of subsets of $K' < K$ points in $\mathbf{t}(K)$. Because this set is much smaller than the original searching space, this pruning step has a low complexity of $O(K^3)$. Finally, they define a heuristic for choosing K based on the magnitude of the increments of the empirical risk when a change point is added.

11.4.2 Hidden Markov Models

Hidden Markov Models (HMMs) assume that the observed copy numbers $\mathbf{c} = (c_j)_{j\in\mathscr{J}}$ are emitted by an underlying Markov chain according to H hidden region-level true copy number states $\boldsymbol{\Gamma} = \{\Gamma_1, \ldots, \Gamma_H\}$. A HMM of order 1 is defined by a specific set $\boldsymbol{\Gamma}$ of hidden states, and transition probabilities $(p(u, v))$ for $(u, v) \in \{1, \ldots, H\}^2$, such that

$$\mathbb{P}(\gamma_{j+1} = \Gamma_v|\gamma_j = \Gamma_u) = p(u, v)\,; \ \forall j \in \{1, \ldots, J\}. \tag{11.13}$$

HMM naturally incorporate and take advantage of the fact that different regions can have the same true copy number, which is not the case of change-point models as the one described by Eq. 11.9. Several HMM-based methods have been proposed for estimating *total* copy numbers. These methods mainly differ in the assumptions that are made for the dynamics of the underlying Markov chain, and the approaches used for the estimation of the hidden states.

The earliest approach assumes that the state sequence is a discrete Markov chain [55]. The number of hidden states is estimated using model selection. More recently, a Bayesian HMM approach with four ($H = 4$) hidden states has been proposed [56]. Because it relies on Bayesian estimation procedures, it provides not only a segmentation of the original observations but also confidence intervals for (the index location of) each copy number change point. However, because the posterior distribution is analytically intractable, posterior inference in this model is performed using simulation-based methods.

The underlying copy number process can also be modeled as a continuous-valued Markov jump process [57]. This type of model is appealing for applications to tumor samples as it does not require the number of hidden states (H) to be specified in advance. Moreover, contrary to [56], the posterior distribution of the hidden variables in [57] can be computed explicitly, which implies that posterior estimates, including confidence assessment of a given segmentation, are available without simulations.

In contrast to change-point methods, HMM-based approaches rely on assumptions on the distribution of the underlying copy number state sequence, and the distribution of the size of copy number regions. Although such assumptions may be unrealistic in the context of cancer studies, a number of state of the art methods for estimating copy numbers from SNP array data use HMMs, as will be explained in Sect. 11.6.

11.5 Purity and Ploidy

Figure 11.4 displays the same data as in Fig. 11.1 after segmentation of total copy numbers by the Circular Binary Segmentation algorithm [38, 42], and estimation of total, minor and major copy numbers as well as allelic ratios in regions of constant total copy numbers.

As explained in Sect. 11.2, TCGA has shown that the copy number states observed in the genomic region displayed in Fig. 11.1 are a normal diploid region $(1, 1)$, a single gain $(1, 2)$ and a copy-neutral LOH $(0, 2)$.

However, it is not straightforward to infer these copy number states only by looking at Fig. 11.4: the observed region-level copy number estimates do not reflect the true copy numbers in the tumor cells of the sample. First, the total copy number is slightly greater than 2 in the normal diploid region. Then, the difference between successive region-level total copy numbers is substantially smaller than the true difference (one copy number unit). Even more strikingly, allele B fractions in the region of copy-neutral LOH (rightmost region) are far from the expected values of 0 or 1.

In this section we explain that these observations are not due to imperfections of the preprocessing method or the microarray assay itself, as they reflect two *biological features* of the data: the ploidy of the tumor, and the presence of normal cells (and possibility of several cytogenetically distinct kinds of tumor cells) in what is usually called a *tumor sample*. For simplicity we will assume that the reference used in the estimation of locus-specific copy numbers (as explained in Sect. 11.3.2) is a cytogenetically normal sample (either normal tissue, or normal blood extract) from the same individual as the tumor. Methods that take these biological parameters into account in the estimation of copy number states are discussed in Sect. 11.6.

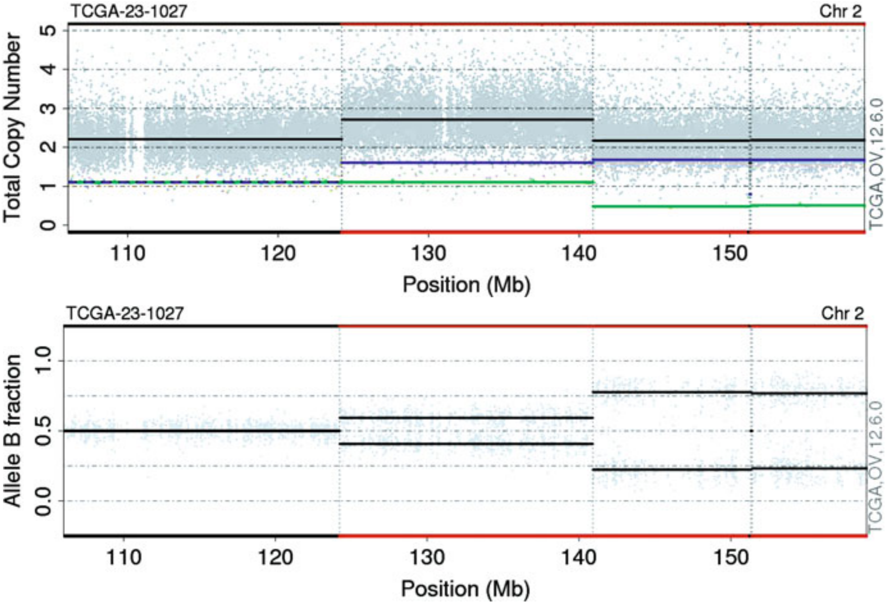


Fig. 11.4 Locus and region-level estimates. Input data is the same as in the top panels of Fig. 11.1. Two main change points in total copy numbers (*top panel*) have been detected by the CBS algorithm [38, 42], and are reported in both panels as dashed gray vertical lines. *Top panel*: locus-level total copy number estimates (*gray dots*), and total (*black*), major (*blue*) and minor (*green*) region-level copy number estimates after change point detection. *Bottom panel*: locus-level (*gray dots*) and region-level allele *B* fractions estimates after change point detection (*black lines*) for heterozygous SNPs. Regions of allelic imbalance (unequal parental copy numbers) are highlighted in red

11.5.1 Pure Tumor Samples

In Fig. 11.5 we have represented the true copy numbers in a sample assumed to contain only one kind of tumor cells and having the same copy number states as those observed in Fig. 11.4: a normal $(1, 1)$ region, followed by a region of gain of a single copy $(1, 2)$, and by a region of copy-neutral LOH $(0, 2)$.

By Eq. 11.2, true allele B fractions satisfy $\beta_j \in \left\{0, \underline{\gamma}_j/\gamma_j, \overline{\gamma}_j/\gamma_j, 1\right\}$, and the pattern of allelic ratios observed in Fig. 11.5 can be interpreted as follows. In a region of allelic balance (left region), where the two parental copy numbers are identical (and not zero), the two heterozygous states merge into $\beta_j = 1/2$ and there are three distinct states: $\beta_j \in \{0, 1/2, 1\}$. In a region of allelic imbalance with retention of heterozygosity (middle region) where the two parental copy numbers are different and neither are zero, β_j can take four distinct values: $\beta_j \in \{0, 1/3, 2/3, 1\}$ for a gain of a single copy of DNA. In a region of LOH (right region), where the minor copy number is 0, heterozygous states disappear and we observe two distinct states: $\beta_j \in \{0, 1\}$. The only type of scenario not represented in Fig. 11.5 is the case of

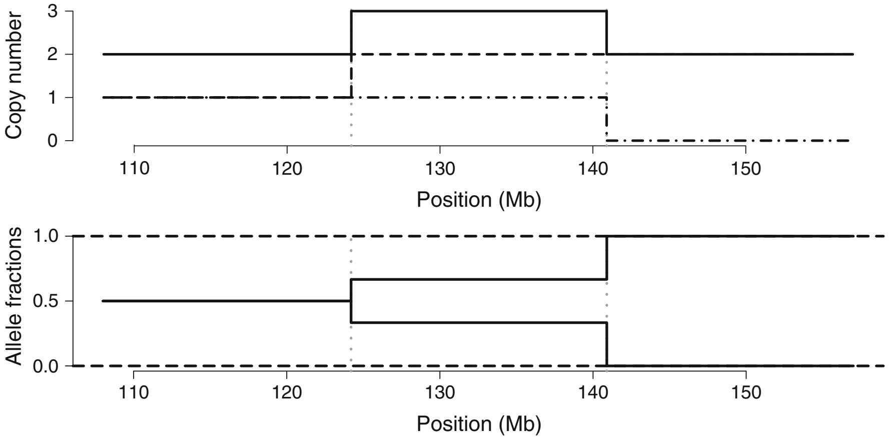


Fig. 11.5 Assumed true total copy numbers and allelic rations in the tumor cells depicted in Figure 11.1. *Top panel*: total (*solid*), major (*dashed*) and minor (*dot-dashed*) copy numbers. *Bottom panel*: allele B fractions: homozygous SNPs (*dashed*) and heterozygous SNPs (*solid*)

homozygous deletions, where both parental copy numbers are null and true allele B fractions are not defined.

11.5.2 Contamination by Normal Cells

In practice however, “tumor samples” are generally a mixture of a tumor cells and a normal cells. In this situation, Eq. 11.2 still holds, but the observed parental copy numbers need not be whole numbers anymore. They are a mixture of the unknown parental copy numbers in the tumor, and the parental copy numbers in normal cells, which are typically but not always $(1, 1)$. The exceptions are so-called copy number polymorphisms (CNPs) [58–60]. For simplicity, we will in what follows only consider SNPs that are diploid in the normal cells.

Assuming that normal cells are diploid, and denoting by $\kappa \in [0, 1]$ the proportion of normal cells in the sample, then the true minor and major copy numbers in the sample are given by

$$\begin{cases} \underline{\gamma}_j &= (1-\kappa)\underline{\gamma}_j^\star + \kappa \\ \overline{\gamma}_j &= (1-\kappa)\overline{\gamma}_j^\star + \kappa \end{cases} \tag{11.14}$$

where $\underline{\gamma}_j^\star$ and $\overline{\gamma}_j^\star$ are the true minor and major copy numbers *of the tumor cells from the sample* at locus j, as if there were no normal cells. Note that these true copy numbers need not be whole numbers either, as the tumor cells of a DNA sample may themselves be a mixture of several tumoral populations (or clones), each with

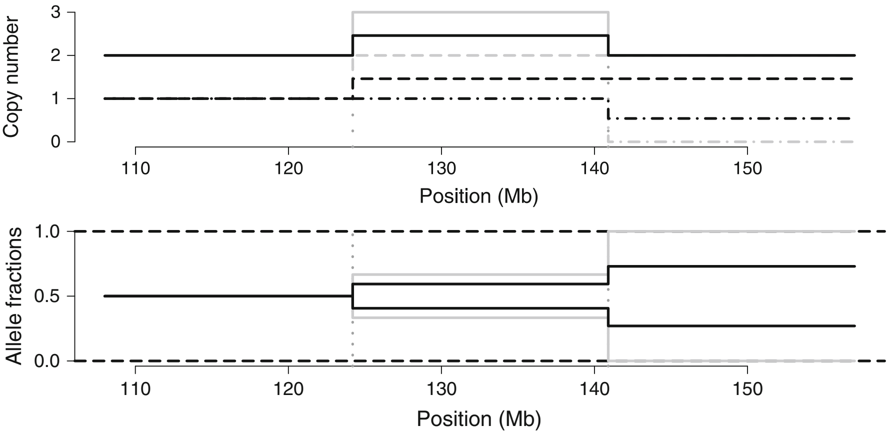


Fig. 11.6 Influence of contamination by normal cells on true copy numbers: comparing 0% contamination (pure tumor as in Figure 11.5, *gray lines*) with 54% contamination (*black lines*). *Top panel*: true total (*solid*), major (*dashed*) and minor (*dot-dashed*) copy numbers. *Bottom panel*: true allele B fractions: homozygous SNPs (*dashed*) and heterozygous SNPs (*solid*)

distinct whole-number copy number profiles. The corresponding total copy numbers and allelic imbalances (when j is a heterozygous SNP) are given by

$$\begin{cases} \gamma_j & = (1-\kappa)\gamma_j^\star + 2\kappa \\ \delta_j & = \frac{\overline{\gamma}_j^\star - \underline{\gamma}_j^\star}{\gamma_j^\star + 2\kappa/(1-\kappa)} \end{cases} \tag{11.15}$$

True allele B fractions satisfy $\beta_j \in \{0, 1/2 - \delta_j/2, 1/2 + \delta_j/2, 1\}$. The influence of normal contamination on true total copy numbers and allelic ratios is shown in Fig. 11.6. Normal contamination moves the observed allelic ratios towards those of the corresponding normal genotypes, and the observed total copy numbers towards the copy number of normal cells. A major difference with the case of no normal contamination is that one still observes heterozygous states in regions of LOH in the tumor: indeed, in regions of LOH, the minor copy number is 0 in tumor cells ($\underline{\gamma}^\star = 0$), and we have

$$\beta \in \left\{0; \frac{\kappa}{(1-\kappa)\gamma^\star + 2\kappa}; \frac{(1-\kappa)\gamma^\star + \kappa}{(1-\kappa)\gamma^\star + 2\kappa}; 1\right\}, \tag{11.16}$$

which corresponds to four distinct modes for allelic ratios. This is illustrated by Fig. 11.6 (right) in the particular situation of copy-neutral LOH, where $\gamma^\star = 2$, leading to $\beta \in \{0; \kappa/2; 1-\kappa/2; 1\}$.

From a modeling point of view, it is worth noting that normal cell contamination is a particular case of contamination, because it may be estimated and corrected for based on either diploid assumptions or explicit measurements of a matched normal (germline) sample. Simply speaking, it is in many cases possible to remove the normal component in the tumor-normal mixture. This is rarely possible for other types of cell contaminations, as they are generally not directly measured. In particular, the problem of identifying different tumor clones from one heterogeneous tumor sample is a harder one.

11.5.3 Tumor Ploidy

As explained in Sect. 11.3.2, the total copy number at locus j is generally estimated relative to a reference as in Eq. 11.5, in order to cancel locus-specific affinities. We can actually interpret c as an estimator of the true copy number in the tumor sample if *the same number of cells were hybridized to the microarray in the tumor and in the normal assay*.

This assumption does not necessarily hold, because of copy number alterations in the tumor. Indeed, the experimental protocol constrains the amount of DNA, not the number of cells, to be the same for each sample assayed [11, 36]. For example, a purely tetraploid tumor with two copies of the genome and no other chromosomal alteration could not be distinguished from a cytogenetically normal (*diploid*) sample, as the genomic material hybridized on the SNP array is the same in both situations. We refer to [35, Sect. 4.4] for further discussion on this issue.

In this chapter we define the *ploidy* λ of a biological sample as the total amount of genomic DNA in this sample relative to that of a normal sample. Therefore, ploidy as defined here needs not be a whole number, because of chromosomal gains and losses and as the tumor sample may be a mixture of normal cells and tumor cells or one or more types of tumor cells with different patterns of genomic alteration. Figure 11.7 illustrates the influence of tumor ploidy on SNP array signals when using Eq. 11.5 to estimate total copy numbers, that is, when assuming that the average true copy number in the normal is 2. Ploidy acts as a scaling factor for total, minor and major copy numbers. Allelic signals as defined in Eq. 11.7 are not affected.

11.5.4 Combined Influence of Purity and Ploidy

As a result of the combined influence of purity and ploidy on the actual composition of a biological sample, the true minor and major copy numbers at a SNP j may be written as

$$\underline{\gamma}_j = \frac{1}{\lambda}\left[(1-\kappa)\underline{\gamma}_j^{\star} + \kappa\right] \tag{11.17}$$

$$\overline{\gamma}_j = \frac{1}{\lambda}\left[(1-\kappa)\overline{\gamma}_j^{\star} + \kappa\right] \tag{11.18}$$

The corresponding true total copy numbers and allelic ratios are given by:

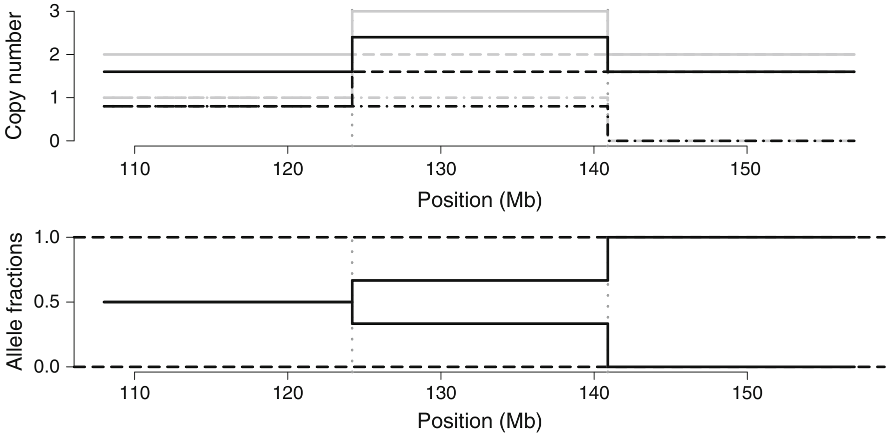


Fig. 11.7 Influence of tumor ploidy on true copy numbers in absence of normal contamination: comparing ploidy 2 (as in Figure 11.5, *gray lines*) to ploidy 2.5 (*black lines*). *Top panel*: true total (*solid*), major (*dashed*) and minor (*dot-dashed*). *Bottom panel*: true allele B fractions: homozygous SNPs (*dashed*) and heterozygous SNPs (*solid*)

$$\gamma_j = \frac{1}{\lambda}\left[(1-\kappa)\gamma_j^\star + 2\kappa\right] \tag{11.19}$$

$$\delta_j = \frac{\overline{\gamma}_j^\star - \underline{\gamma}_j^\star}{\gamma_j^\star + 2\kappa/(1-\kappa)} \tag{11.20}$$

As explained above, we note that allelic imbalances (δ_j) are only affected by normal contamination, not by ploidy. Figure 11.8 illustrates the combined influence of purity and ploidy by comparing the true total copy numbers and allelic ratios for a pure tumor without normal contamination (as in Fig. 11.5) with a non-diploid tumor with normal contamination according to Eqs. 11.19 and 11.20.

When accounting for both purity and ploidy, the copy number patterns become quite similar to those observed with real data. This is illustrated by the comparison between the true copy numbers in Fig. 11.8 and locus- and region-level copy number estimates in Fig. 11.4. For this particular sample, TCGA reported 54% of normal cells and ploidy 1.8; these estimates were used to produce Fig. 11.8.

11.6 Estimation of Copy Number States in Cancer Studies

Copy number studies in cancer research aim at identifying the unknown copy number state in a tumor sample, as defined in Sect. 11.2. As explained above, the word *identification* actually covers two different statistical questions: *detecting* changes in copy number signals, and *calling* regions, that is, assigning a copy number state

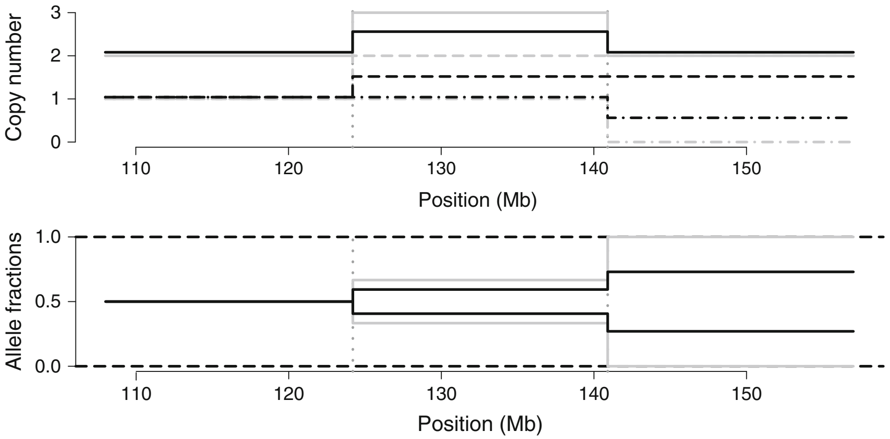


Fig. 11.8 Combined influence of tumor ploidy and normal cell contamination on true copy numbers: comparing ploidy 2 and no normal contamination (as in Fig. 11.5, *gray lines*) with ploidy 1.8 and 54% normal contamination (*black lines*). *Top panel*: true total (*solid*), major (*dashed*) and minor (*dot-dashed*). *Bottom panel*: true allele *B* fractions: homozygous SNPs (*dashed*) and heterozygous SNPs (*solid*)

to each region detected. Because SNP arrays interrogate allele-specific signals, they can be used for both detection and calling.

Segmentation can be performed regardless of purity and ploidy, although these two biological parameters do influence the detection power of any given segmentation method, through the distance between true region-level copy number states. However, both purity and ploidy have to be acknowledged in order to call copy number states in the tumor cells of a given sample.

11.6.1 Existing Methods

A number of methods for analyzing SNP array data were developed in the context of Copy Number Variation (CNV) studies in normal samples: VanillaICE [61], PennCNV [62], QuantiSNP [63], and BirdSuite [64]. Most of them are based on HMMs. Because these methods are dedicated to, and well-designed for CNV studies, their model states do not adequately describe the copy number states in Table 11.1. More specifically, either they do not consider allele-specific amplifications [62, 63], or the distinction between normal and copy-neutral LOH [61], or they are only designed to detect rare CN aberrations [64]. Moreover, their states generally do not account for possible tumor heterogeneity or contamination by normal cells.

Table 11.3 lists methods that actually combine total and allele-specific signals in order to call copy number states (as defined in Table 11.1) in cancer studies. They are described in terms of the type of information they take into account and the type

Table 11.3 Existing methods for copy number studies in cancers using SNP arrays. Settings: has the method been developed for studies with available paired normal samples, or not? Last two columns: does the method explicitly account for ploidy, purity?

Name	Settings	Detection method	Ploidy	Purity
MCP [8]	unpaired	HMM on δ	no	no
Gardina [11]	unpaired	HMM on "genotypes"	yes	no
BAFsegmentation [6]	paired or unpaired	segmentation of δ	no	yes
SOMATICs [7]	unpaired	segmentation of δ	no	yes
AsCNAR/CNAG [36]	unpaired	HMM on δ	no	yes
OverUnder [65]	unpaired	2×1d smoothing	yes	no
PSCBS [66]	paired	two-way segmentation	no	no
GAP [67]	unpaired	2×1d segmentation	yes	yes
Lamy [68]	paired	HMM on (γ, δ)	no	yes
PSCN [69]	unpaired	HMM on (γ, δ)	no	no
PICNIC [43]	unpaired	HMM on (γ, δ)	yes	no
genoCNA [70]	unpaired	HMM on (γ, δ)	no	yes

of method they use for *detecting* copy number changes, whether their application requires the availability of a paired normal reference, and whether they explicitly account for tumor purity and ploidy as discussed in Sect. 11.5.

We have shown in Sect. 11.2 that SNP array signals were two-dimensional by nature, and that both dimensions were needed to *call* copy number states as defined in Table 11.1. All methods cited in Table 11.3 indeed make use of both dimensions at the calling step, but not necessarily at the detection step. These methods can be classified in terms of the type of input data they are using at the detection step, as indicated by the horizontal lines in Table 11.3. We note here that although raw allelic signals typically have several modes in a region of constant copy number (as explained in Sect. 11.2), direct segmentation methods can be used to detect changes in allelic signals from SNPs that are heterozygous in the germline [6, 7, 66, 67].

Methods from the first group use only one piece of information for detection [6–8, 11, 36]. As these methods are mostly interested in loss of heterozygosity, they all take allelic imbalances (or genotypes) and not total copy numbers, as an input for the detection step. Methods from the second group combine both pieces of information, either by independent smoothing [65] or segmentation [67] of each piece of information, or by segmentation of total signals followed by segmentation of allelic signals [66]. In particular, GAP [67] is to our knowledge the only method that explicitly accounts for both purity and ploidy. Finally, methods from the third group perform truly joint detection of copy number changes [43, 68–70]. In the next section, we show that such joint approaches can be more powerful to detect copy number changes.

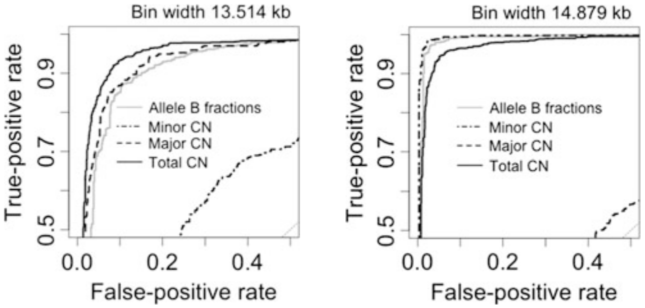


Fig. 11.9 At a fixed resolution, total copy numbers and allelic signals have comparable power to detect copy number changes. ROC curves for the two copy number change points studied in Fig. 11.1: *left panel*, between a normal region (1, 1) and a single gain (1, 2); *right panel*, between a single gain (1, 2) and a region of copy-neutral LOH (0, 2). Affymetrix GenomeWideSNP_6 data

11.6.2 Joint Detection Provides more Power to Detect Copy Number Changes

In this section, we demonstrate that there is substantial statistical power to gain by considering both pieces of information for the detection step. Figure 11.9 shows that total and allelic signals have comparable power to detect the two change points studied throughout this chapter: a transition between a diploid normal state $(1,1)$ and a gain $(1,2)$ (left panel), and a transition between a gain $(1,2)$ and a region of copy-neutral LOH $(0,2)$ (right panel). For each type of signal studied in Fig. 11.9 is a ROC curve used to measure the separation between two copy number states at a change point of known location based on this signal. We refer to [14, 15, 41] for a comprehensive description of this evaluation.

Allelic signals have a lower density than total signals as copy number probes only measure total copy numbers, and because only SNPs that are heterozygous in the germline are informative in terms of allelic imbalances. However, these ROC curves can be compared across signals because the evaluation is performed at a fixed *resolution* for each change point. Each resolution corresponds to a different number of markers for allelic and total signals. The change point between states $(1,1)$ and $(1,2)$ is detected slightly better with total signals (*solid black line*) than with allelic signals: allelic imbalances (*solid gray*) or major (*dashed*) copy numbers. As expected, the change point is not detected by minor copy numbers (*dot-dashed*), as there is no change in true minor copy numbers. The change point between $(1,2)$ and $(0,2)$ is detected with similar or higher power using allelic signals than using total signals. Similar patterns are observed for other types of change points, suggesting that there is substantial detection power to gain in using both total and allelic signals for the detection of copy number changes.

11.6.3 Comparison Between Existing Joint Methods

Four methods based on HMM perform truly joint TCN and AI analyses: PICNIC [43], PSCN [69], genoCNA [70], and the method proposed in [68]. One advantage of HMM-based methods is that they can incorporate different probe types (SNPs and copy number probes) naturally, although in practice this seems to have been done only in PICNIC [43].

As discussed in Sect. 11.4, HMMs with discrete hidden state spaces perform the detection and calling steps at the same time, and are necessarily limited in terms of number of copy number states, that is, they cannot adapt to the intrinsic number of copy number states of a given problem. To our knowledge, PSCN [69] is currently the only method for joint TCN and AI analysis which is based on a continuous hidden state space. Conversely, one drawback of this type of approach is that it does not give a hard segmentation of the data in copy number states. Instead, copy numbers are estimated at each particular location and the method has to be combined with some thresholding in order to actually provide a segmentation of the original data. Moreover, downstream analyses are needed to estimate and/or call minor and major copy numbers.

We advocate the development of a joint direct segmentation method, that could take fully advantage of the two dimensions of SNP array data, as the above HMM do, but without assuming a particular form for the distribution of the copy number states sequence or the distribution of the size of copy number regions. Such a method could rely on the same type of models as those developed for joint direct segmentation of several copy number profiles [72–74].

11.7 Concluding Remarks

In this chapter, we have underlined key aspects the analysis of SNP array data, including the influence of purity and ploidy on the observed data, and explained how they should be accounted for in the identification of copy number states. Although existing methods adequately address several of the challenges we focused on in this chapter, a few questions remain to be solved besides the above-mentioned development of a joint direct segmentation method. For the problem of detecting copy number changes, most existing methods assume that the errors follow a Gaussian distribution, although microarray data may be more heavy tailed. Current statistical models can be extended to other types of error distribution, but the main difficulty resides in developing efficient practical implementations.

For calling copy number states, although the effects of purity and ploidy are now widely acknowledged, methods to account for them – and also for tumor heterogeneity, that is, the possible presence of several tumoral clones in the tumor sample – will probably have to be improved and adapted to different types of cancers. A critical assessment of such methods is desirable, and would require producing validation data where purity and ploidy are known.

We have focused on the identification of copy number changes for one sample from one SNP array platform. In conclusion, we indicate statistical questions that arise in more general settings: when several samples are considered at a time, when one sample has been assayed on several platforms, and with newer copy number technologies.

11.7.1 Identifying Recurrent Allele-Specific Events

Even though some of the preprocessing methods described in Sect. 11.3 require several microarrays, currently available methods for identifying copy number states from SNP arrays analyze each tumor sample separately. However, the joint analysis of several samples from the same tumor type should be more powerful if the same biological events can be shared by several samples, as already demonstrated for total copy numbers for array-CGH data [72–75]. Extensions of such methods to allelic signals remain to be developed.

11.7.2 Combining Allele-Specific Signals Across Platforms

When the same sample is analyzed by two different platforms, combining signals across platforms should lead to improved detection of copy number alterations. This has been demonstrated for total copy numbers [41,76] but still has to be investigated for allelic signals.

11.7.3 High-Throughput Sequencing

Currently, high-throughput sequencing technologies are more expensive than SNP arrays for whole genome allele-specific copy number studies, because accurate estimation of allelic ratios from read count data requires high sequencing coverage. The rapid evolution of these technologies suggests that allele-specific copy number studies will be cost-effective in the near future, leading to new statistical issues that will need to be addressed.

Acknowledgements We gratefully acknowledge the Lawrence Berkeley National Laboratory (LBNL) and The Cancer Genome Atlas (TCGA) for making data and results available. This work was supported by NCI grant U24 CA126551.

References

1. Hanahan, D., & Weinberg, R. A. (2000, January). The hallmarks of cancer. *Cell*, *100*(1), 57–70.
2. Chin, L., & Gray, J. W. (2008, April). Translating insights from the cancer genome into clinical practice. *Nature*, *452*(7187), 553–563.

3. Pinkel, D., Segraves, R., Sudar, D., Clark, S., Poole, I., Kowbel, D., Collins, C., Kuo, W. L., Chen, C., Zhai, Y., Dairkee, S. H., Ljung, B. M., Gray, J. W., & Albertson, D. G. (1998). High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. *Nature Genetics*, *20*, 207–211.
4. Albertson, D. G., & Pinkel, D. (2003, October). Genomic microarrays in human genetic disease and cancer. *Human Molecular Genetics*, *12*(Spec. No. 2), R145–R152.
5. Tuna, M., Knuutila, S., & Mills, G. B. (2009, March). Uniparental disomy in cancer. *Trends in Molecular Medicine*, *15*(3), 120–128. PMID: 19246245.
6. Staaf, J., Lindgren, D., Vallon-Christersson, J., Isaksson, A., Goransson, H., Juliusson, G., Rosenquist, R., Hoglund, M., Borg, A., & Ringner, M. (2008). Segmentation-based detection of allelic imbalance and loss-of-heterozygosity in cancer cells using whole genome SNP arrays. *Genome Biology*, *9*(9), R136.
7. Assié, G., LaFramboise, T., Platzer, P., Bertherat, J., Stratakis, C. A., & Eng, C. (2008). SNP arrays in heterogeneous tissue: Highly accurate collection of both germline and somatic genetic information from unpaired single tumor samples. *American Journal of Human Genetics*, *82*, 903–915.
8. Li, C., Beroukhim, R., Weir, B. A., Winckler, W., Garraway, L. A., Sellers, W. R., & Meyerson, M. (2008). Major copy proportion analysis of tumor samples using SNP arrays. *BMC Bioinformatics*, *9*, 204.
9. LaFramboise, T. (2009, July). Single nucleotide polymorphism arrays: A decade of biological, computational and technological advances. *Nucleic Acids Research*, *37*(13), 4181–4193. PMID: 19570852.
10. Peiffer, D. A., Le, J. M., Steemers, F. J., Chang, W., Jenniges, T., Garcia, F., Haden, K., Li, J., Shaw, C. A., Belmont, J., Cheung, S. W., Shen, R. M., Barker, D. L., & Gunderson, K. L. (2006, September). High-resolution genomic profiling of chromosomal aberrations using Infinium whole-genome genotyping. *Genome Research*, *16*(9), 1136–1148.
11. Gardina, P. J., Lo, K. C., Lee, W., Cowell, J. K., & Turpaz, Y. (2008). Ploidy status and copy number aberrations in primary glioblastomas defined by integrated analysis of allelic ratios, signal ratios and loss of heterozygosity using 500 K SNP Mapping Arrays. *BMC Genomics*, *9*(1), 489.
12. Collins, F. S., & Barker, A. D. (2007, March). Mapping the cancer genome. *Scientific American*, *296*(3), 50–57.
13. The Cancer Genome Atlas (TGCA) research Network. (2008). Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, *455*, 1061–1068.
14. Bengtsson, H., Wirapati, P., & Speed, T. P. (2009). A single-array preprocessing method for estimating full-resolution raw copy numbers from all Affymetrix arrays including GenomeWideSNP 5 & 6. *Bioinformatics*, *27*(17), 2149–2156.
15. Bengtsson, H., Neuvial, P., & Speed, T. P. (2010). TumorBoost: Normalization of allele-specific tumor copy numbers from a single pair of tumor-normal genotyping microarrays. *BMC Bioinformatics*, *11*(1), 245.
16. Affymetrix Inc. (2007). Affymetrix Genome-Wide Human SNP Array 6.0. Data sheet.
17. Affymetrix Inc. (2009). Affymetrix cytogenetics research solution. Data sheet.
18. Gunderson, K. L., Steemers, F. J., Lee, G., Mendoza, L. G., & Chee, M. S. (2005, May). A genome-wide scalable SNP genotyping assay using microarray technology. *Nature Genetics*, *37*(5), 549–554.
19. Steemers, F. J., & Gunderson, K. L. (2007). Whole genome genotyping technologies on the BeadArray platform. *Biotechnology Journal*, *2*(1), 41–49.
20. Illumina, Inc. (2009). SNP genotyping and copy number analysis. Illumina Product Guide.
21. Nannya, Y., Sanada, M., Nakazaki, K., Hosoya, N., Wang, L., Hangaishi, A., Kurokawa, M., Chiba, S., Bailey, D. K., Kennedy, G. C., & Ogawa, S. (2005, July 15). A robust algorithm for copy number detection using high-density oligonucleotide single nucleotide polymorphism genotyping arrays. *Cancer Research*, *65*(14), 6071–6079.
22. Ishikawa, S., Komura, D., Tsuji, S., Nishimura, K., Yamamoto, S., Panda, B., Huang, J., Fukayama, M., Jones, K. W., & Aburatani, H. (2005, August 12). Allelic dosage analysis with genotyping microarrays. *Biochemical and Biophysical Research Communications*, *333*(4), 1309–1314.

23. Carvalho, B., Bengtsson, H., Speed, T. P., & Irizarry, R. A. (2007, April). Exploration, normalization, and genotype calls of high-density oligonucleotide SNP array data. *Biostatistics*, *8*(2), 485–499.
24. Bengtsson, H., Irizarry, R., Carvalho, B., & Speed, T. P. (2008, March 15). Estimation and assessment of raw copy numbers at the single locus level. *Bioinformatics*, *24*(6), 759–767.
25. Yang, Y. H., Dudoit, S., Luu, P., Lin, D. M., Peng, V., Ngai, J., & Speed, T. P. (2002). Normalization for cDNA microarray data: A robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research*, *30*(4), e15.
26. Li, C., & Wong, W. H. (2001, January 2). Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(1), 31–36.
27. Bolstad, B. M., Irizarry, R. A., Astrand, M., & Speed, T. P. (2003, January). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, *19*(2), 185–193.
28. Ortiz-Estevez, M., Bengtsson, H., & Rubio, A. (2010, June). ACNE: A summarization method to estimate allele-specific copy numbers for Affymetrix SNP arrays. *Bioinformatics*, *26*(15), 1827–1833.
29. Rabbee, N., & Speed, T. P. (2006, January). A genotype calling algorithm for Affymetrix SNP arrays. *Bioinformatics*, *22*(1), 7–12.
30. Affymetrix Inc. (2006, April). BRLMM: An improved genotype calling method for the GeneChip Human Mapping 500 K Array Set.
31. LaFramboise, T., Harrington, D., & Weir, B. A. (2007, April). PLASQ: A generalized linear model-based procedure to determine allelic dosage in cancer cells from SNP array data. *Biostatistics*, *8*(2), 323–336.
32. Illumina, Inc. (2006). Illumina's genotyping data normalization methods. White paper.
33. Staaf, J., Vallon-Christersson, J., Lindgren, D., Juliusson, G., Rosenquist, R., Hoglund, M., Borg, A., & Ringner, M. (2008). Normalization of illumina infinium whole-genome SNP data improves copy number estimates and allelic intensity ratios. *BMC Bioinformatics*, *9*(1), 409.
34. Steemers, F. J., Chang, W., Lee, G., Barker, D. L., Shen, R., & Gunderson, K. L. (2006). Whole-genome genotyping with the single-base extension assay. *Nature Methods*, *3*(1), 31–33. PMID: 16369550.
35. Bengtsson, H. (2004, October). *Low-level analysis of microarray data*. PhD thesis, Centre for Mathematical Sciences, Division of Mathematical Statistics, Lund University. http://www.lunduniversity.lu.se/o.o.i.s?id=24732&postid=467374
36. Yamamoto, G., Nannya, Y., Kato, M., Sanada, M., Levine, R. L., Kawamata, N., Hangaishi, A., Kurokawa, M., Chiba, S., Gilliland, D. G., Koeffler, H. P., & Ogawa, S. (2007, July). Highly sensitive method for genomewide detection of allelic composition in nonpaired, primary tumor specimens by use of affymetrix single-nucleotide-polymorphism genotyping microarrays. *American Journal of Human Genetics*, *81*(1), 114–126.
37. Pounds, S., Cheng, C., Mullighan, C., Raimondi, S. C., Shurtleff, S., & Downing, J. R. (2009). Reference alignment of SNP microarray signals for copy number analysis of tumors. *Bioinformatics*, *25*(3), 315.
38. Olshen, A. B., Venkatraman, E. S., Lucito, R., & Wigler, M. (2004). Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, *5*(4), 557–572.
39. Bengtsson, H., Simpson, K., Bullard, J., & Hansen, K. (2008). *aroma. affymetrix: A generic framework in R for analyzing small to very large Affymetrix data sets in bounded memory* Technical Report 745. Berkeley: Department of Statistics, University of California.
40. von Neumann, J., Kent, R. H., Bellinson, H. R., & Hart, B. I. (1941). The mean square successive difference. *The Annals of Mathematical Statistics*, *12*(2), 153–162.
41. Bengtsson, H., Ray, A., Spellman, P. T., & Speed, T. P. (2009). A single-sample method for normalizing and combining full-resolution copy numbers from multiple sources and technologies. *Bioinformatics*, *25*(7), 861—867.
42. Venkatraman, E. S., & Olshen, A. B. (2007, March). A faster circular binary segmentation algorithm for the analysis of array CGH data. *Bioinformatics*, *23*(6), 657–663.

43. Greenman, C. D., Bignell, G., Butler, A., Edkins, S., Hinton, J., Beare, D., Swamy, S., Santarius, T., Chen, L., Widaa, S., Futreal, P. A., & Stratton, M. R. (2010). PICNIC: An algorithm to predict absolute allelic copy number variation with microarray cancer data. *Biostatistics*, *11*(1), 164–175.
44. Lai, W. R., Johnson, M. D., Kucherlapati, R., & Park, P. J. (2005, October 1). Comparative analysis of algorithms for identifying amplifications and deletions in array-CGH data. *Bioinformatics*, *21*(19), 3763–3770.
45. Willenbrock, H., & Fridlyand, J. (2005, November 15). A comparison study: Applying segmentation to array-CGH data for downstream analyses. *Bioinformatics*, *21*(22), 4084–4091.
46. Jong, K., Marchiori, E., van der Vaart, A., Ylstra, B., Weiss, M., & Meijer, G. (2003, April 14–16). Chromosomal breakpoint detection in human cancer. In G. R. Raidl, S. Cagnoni, J. J. R. Cardalda, D. W. Corne, J. Gottlieb, A. Guillot, E. Hart, C. G. Johnson, E. Marchiori, J.-A. Meyer, & M. Middendorf (Eds.), *Applications of evolutionary computing, EvoWorkshops2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, EvoSTIM*, Vol. 2611 of *LNCS* (pp. 54–65). England, UK: University of Essex, Springer-Verlag.
47. Zhang, N. R., & Siegmund, D. O. (2007). A modified bayes information criterion with applications to the analysis of comparative genomic hybridization data. *Biometrics*, *63*(1), 22–32.
48. Lavielle, M. (2005). Using penalized contrasts for the change-point problem. *Signal Processing*, *85*(8), 1501–1510.
49. Picard, F., Robin, S., Lavielle, M., Vaisse, C., & Daudin, J. J. (2005). A statistical approach for array CGH data analysis. *BMC Bioinformatics*, *6*(1), 27–27.
50. Rigaill, G. (2010, April). Pruned dynamic programming for optimal multiple change-point detection. Arxiv preprint arXiv:1004.0887.
51. Tibshirani, R., & Wang, P. (2008, Jan). Spatial smoothing and hot spot detection for CGH data using the fused lasso. *Biostatistics*, *9*(1), 18–29.
52. Harchaoui, Z., & Lévy-Leduc, C. (2008). Catching change-points with lasso. *Advances in Neural Information Processing Systems*, *20*, 161–168.
53. Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., & Knight, K. (2005). Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, *67*(1), 91–108.
54. Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *Annals of statistics*, *32*(2), 407–451.
55. Fridlyand, J., Snijders, A., Pinkel, D., Albertson, D. G., & Jain, A. N. (2004). Application of hidden markov models to the analysis of the array CGH data. *Journal of Multivariate Analysis*, *90*, 132–153. Special Issue on Multivariate Methods in Genomic Data Analysis.
56. Guha, S., Li, Y., & Neuberg, D. (2008). Bayesian hidden Markov modeling of array CGH data. *Journal of the American Statistical Association*, *103*(482), 485–497.
57. Lai, T. L., Xing, H., & Zhang, N. (2008, April). Stochastic segmentation models for array-based comparative genomic hybridization data analysis. *Biostatistics*, *9*(2), 290–307.
58. Sebat, J., Lakshmi, B., Troge, J., Alexander, J., Young, J., Lundin, P., Manér, S., Massa, H., Walker, M., Chi, M., Navin, N., Lucito, R., Healy, J., Hicks, J., Ye, K., Reiner, A., Gilliam, T. C., Trask, B., Patterson, N., Zetterberg, A., & Wigler, M. (2004, July). Large-scale copy number polymorphism in the human genome. *Science*, *305*(5683), 525–528.
59. Iafrate, A. J., Feuk, L., Rivera, M. N., Listewnik, M. L., Donahoe, P. K., Qi, Y., Scherer, S. W., & Lee, C. (2004, September). Detection of large-scale variation in the human genome. *Nature Genetics*, *36*(9), 949–951.
60. Redon, R., Ishikawa, S., Fitch, K. R., Feuk, L., Perry, G. H., Andrews, T. D., et al. (2006). Global variation in copy number in the human genome. *Nature*, *444*, 444–454.
61. Scharpf, R. B., Parmigiani, G., Pevsner, J., & Ruczinski, I. (2008). Hidden Markov models for the assessment of chromosomal alterations using high-throughput SNP arrays. *Annals of Applied Statistics*, *2*(2), 687–713.
62. Wang, K., Li, M., Hadley, D., Liu, R., Glessner, J., Grant, S. F. A., Hakonarson, H., & Bucan, M. (2007, November). PennCNV: An integrated hidden markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome Research*, *17*(11), 1665.

63. Colella, S., Yau, C., Taylor, J. M., Mirza, G., Butler, H., Clouston, P., Bassett, A. S., Seller, A., Holmes, C. C., & Ragoussis, J. (2007, March). QuantiSNP: An objective bayes Hidden-Markov model to detect and accurately map copy number variation using SNP genotyping data. *Nucleic Acids Research*, *35*(6), 2013–2025.
64. Korn, J. M., Kuruvilla, F. G., McCarroll, S. A., Wysoker, A., Nemesh, J., Cawley, S., Hubbell, E., Veitch, J., Collins, P. J., Darvishi, K., Lee, C., Nizzari, M. M., Gabriel, S. B., Purcell, S., Daly, M. J., & Altshuler, D. (2008, October). Integrated genotype calling and association analysis of SNPs, common copy number polymorphisms and rare CNVs. *Nature Genetics*, *40*(10), 1253–1260.
65. Attiyeh, E. F., Diskin, S. J., Attiyeh, M. A., Mossé, Y. P., Hou, C., Jackson, E. M., Kim, C., Glessner, J., Hakonarson, H., Biegel, J. A., & Maris, J. M. (2009, February). Genomic copy number determination in cancer cells from single nucleotide polymorphism microarrays based on quantitative genotyping corrected for aneuploidy. *Genome Research*, *19*(2), 276–283.
66. Olshen, A. B., Olshen, R. A., Bengtsson, H., Neuvial, P., Spellman, P. T., & Seshan, V. E. (2010, May). Parent-specific copy number in paired tumor-normal studies using circular binary segmentation. Submitted, December 2010.
67. Popova, T., Manié, É., Stoppa-Lyonnet, D., Rigaill, G., Barillot, E., & Stern, M.-H. (2009). Genome alteration print (GAP): A tool to visualize and mine complex cancer genomic profiles obtained by SNP arrays. *Genome Biology*, *10*(11), R128.
68. Lamy, P., Andersen, C. L., Dyrskjot, L., Torring, N., & Wiuf, C. (2007). A Hidden Markov Model to estimate population mixture and allelic copy-numbers in cancers using Affymetrix SNP arrays. *BMC Bioinformatics*, *8*(1), 434.
69. Chen, H., Xing, H., & Zhang, N. R. (2011 Jan). *Estimation of parent specific DNA copy number in tumors using high-density genotyping arrays*. PLoS Comput Biol., *7*(1): e1001060.
70. Sun, W., Wright, F. A., Tang, Z., Nordgard, S. H., Van Loo, P., Yu, T., Kristensen, V. N., & Perou, C. M. (2009, September). Integrated study of copy number states and genotype calls using high-density SNP arrays. *Nucleic Acids Research*, *37*(16), 5365–5377.
71. Beroukhim, R., Lin, M., Park, Y., Hao, K., Zhao, X., Garraway, L. A., Fox, E. A., Hochberg, E. P., Mellinghoff, I. K., Hofer, M. D., Descazeaud, A., Rubin, M. A., Meyerson, M., Wong, W. H., Sellers, W. R., & Li, C. (2006, May). Inferring loss-of-heterozygosity from unpaired tumors using high-density oligonucleotide SNP arrays. *PLoS Computational Biology*, *2*(5), e41.
72. Zhang, N. R., Siegmund, D. O., Ji, H., & Li, J. Z. (2010). Detecting simultaneous change-points in multiple sequences. *Biometrika*, *97*(3), 631–645.
73. Vert J.-P. & Bleakley K. (2010). Fast detection of multiple change-points shared by many signals using group LARS. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, (eds.) *Advances in Neural Information Processing Systems 23 (NIPS)*, 2343–2351.
74. Picard, F., Lebarbier, É., Budinaská, E., & Robin, S. (2011). *Joint segmentation of multivariate Gaussian Processes using mixed linear models*. Computational Statistics and Data Analysis, *55*, 1160–1170.
75. Shah, S. P., Lam, W. L., Ng, R. T., & Murphy, K. P. (2007, July). Modeling recurrent DNA copy number alterations in array-CGH data. *Bioinformatics*, *23*(13), i450–i458.
76. Zhang, N. R., Senbabaoglu, Y., & Li, J. Z. (2009, November). Joint estimation of DNA copy number from multiple platforms. *Bioinformatics*, *26*(2), 153–160.

Chapter 12
Computational Analysis of ChIP-chip Data

Hongkai Ji

Abstract Chromatin immunoprecipitation coupled with genome tiling array hybridization, also known as ChIP-chip, is a powerful technology to identify protein-DNA interactions in genomes. It is widely used to locate transcription factor binding sites and histone modifications. Data generated by ChIP-chip provide important information on gene regulation. This chapter reviews fundamental issues in ChIP-chip data analysis. Topics include data preprocessing, background correction, normalization, peak detection and motif analysis. Statistical models and principles that significantly improve data analysis are discussed. Popular software tools are briefly introduced.

12.1 Introduction

ChIP-chip (or ChIP-on-chip) [32] is a recently developed approach to study genome-wide protein-DNA interactions. It has been widely used to locate transcription factor binding sites [7–9] and histone modifications in genomes [5]. The word "ChIP-chip" stands for Chromatin ImmunoPrecipitation (ChIP) followed by DNA microarray (chip) hybridization. The workflow of this technology is illustrated in Fig. 12.1. Briefly, a protein of interest is cross-linked with the DNA (chromatin) it binds to. The cells are lysed and the chromatin is sheared into small fragments either by sonication or by cutting with restriction enzymes. The protein of interest, together with the bound chromatin fragments, are precipitated using a protein-specific antibody. This is a procedure known as chromatin immunoprecipitation (ChIP). Next, chromatin fragments are dissociated from the protein via reverse cross-linking. Fragmented DNA is purified, amplified, denatured and labeled with fluorescent tags, creating a sample called ChIP sample. The length

H. Ji
Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health, 615 North Wolfe Street, Baltimore, MD 21205
e-mail: hji@jhsph.edu

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_12,
© Springer-Verlag Berlin Heidelberg 2011

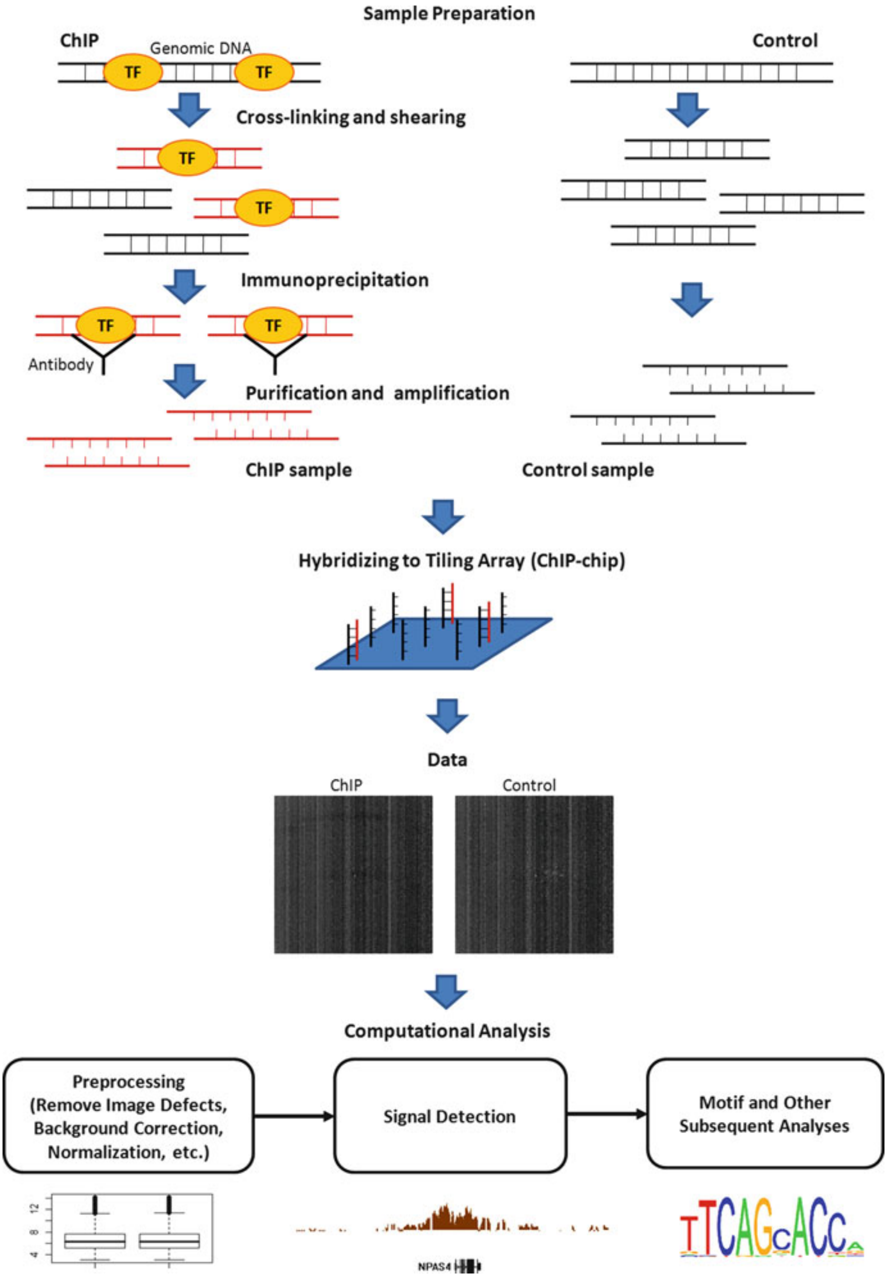

Sample Preparation
ChIP
Genomic DNA
TF
TF
Control
Cross-linking and shearing
TF
TF
Immunoprecipitation
TF
TF
Antibody
Purification and amplification
ChIP sample
Control sample
Hybridizing to Tiling Array (ChIP-chip)
Data
ChIP
Control
Computational Analysis
Preprocessing (Remove Image Defects, Background Correction, Normalization, etc.)
Signal Detection
Motif and Other Subsequent Analyses
12
8
4
NPAS4

Fig. 12.1 Workflow of ChIP-chip

of DNA fragments in the ChIP sample ranges from 500 to 1,000 base pairs (bp). Control samples are prepared by skipping the immunoprecipitation or replacing the protein-specific antibody with a non-specific one. Compared to control samples, ChIP samples are enriched in DNA fragments bound by the protein of interest. Both ChIP and control samples can then be hybridized to tiling arrays. These are specially designed microarrays that use 25–60 bp long oligonucleotide probes to measure the abundance of particular molecules in a DNA sample. The probes are selected from a reference genome assembly to cover the entire or targeted regions of the genome, with an average probe spacing (i.e. distance between two neighboring probes) ranging from a few to a few hundred base pairs. After hybridizing ChIP and control samples to tiling arrays, locations in the genome that are bound by the protein of interest are highlighted by contiguous stretches of probes for which fluorescence intensities in ChIP samples are significantly higher than intensities in control samples. Using this technology, one can map transcription factor binding sites and histone modifications in complex genomes in a more unbiased manner. ChIP-chip data collected at various time points and in various cell types provide critical information for unraveling the complex gene regulatory programs in the genome. Eventually, this knowledge will help us understand human diseases better and find better disease treatment strategies.

Data produced by ChIP-chip experiments are vast and noisy. Extracting meaningful information from these data requires a multiple-step procedure. First, raw intensities produced by array hybridization need to be extracted, outliers and systematic biases need to be removed, and probes in the array need to be mapped back to the genome. This procedure is referred to as data preprocessing. Second, locations of protein-DNA interactions need to be detected by separating biologically relevant signals from noise. This procedure is called “peak detection” or “signal detection”. Third, if an experiment studies transcription factor (TF) binding, there is a need to identify DNA sequence motifs that are recognized by the TF. Information on binding motifs is useful for subsequent experiments. For example, they can be used to design knock-out experiments to verify functions of a *cis*-regulatory element. In this chapter, methods dealing with these various topics will be reviewed. In particular, Sect. 12.2 provides an overview of data preprocessing. A detailed discussion on two topics of preprocessing, normalization and background correction, will be given in Sect. 12.3. Section 12.4 introduces methods for signal detection (i.e. detecting protein-DNA interactions). Methods for identifying transcription factor binding motifs are discussed in Sect. 12.5. Section 12.6 will conclude the Chapter by discussing several open issues and challenges in the ChIP-chip data analysis.

To date, three major tiling array platforms are widely used in ChIP-chip studies: Affymetrix, NimbleGen, and Agilent. The Affymetrix arrays use 25 bp oligonucleotides as probes. Each array can contain about six million probes with a 35 bp probe spacing (i.e. the average distance between two neighboring probes is 35 bp). The entire human genome can be tiled using seven arrays that contain 42 million probes in total. The NimbleGen arrays use 50–75 bp oligonucleotide probes. Each array contains up to 2.1 million probes. With a 100 bp probe spacing, the entire human genome can be covered by ten arrays. The Agilent arrays use 60 bp long probes. Each array can place 244,000 probes. Both NimbleGen and Agilent allow flexible custom array design, whereas Affymetrix provides the lowest cost per probe and highest genomic resolution [29]. Data processing methods differ between platforms. Our disucssion is mainly focused on processing of the Affymetrix tiling arrays. In spite of this, many statistical principles discussed below are general and applicable to the other array platforms as well.

12.2 Data Preprocessing

Following sample hybridization, the arrays are scanned by laser to excite fluorescence. Fluorescence intensities of all probes are stored as images. After image processing, the intensity value of each probe is summarized by a number. For Affymetrix tiling arrays, these numbers (i.e. probe intensities) are written into CEL files which are used by most computational biologists as the starting point of data analysis. The scanned images sometimes contain visible artifacts (Fig. 12.2). If one is unlucky, these artifacts could involve a significant proportion of probes. As they may seriously affect signal detection, data analysts are advised to examine the scanned images before carrying out any analyses. If significant artifacts exist, one should remove or exclude them from subsequent analyses. To visually examine the images, one can either use the vendor provided software tools that come with the scanner, or use the free and open source software CisGenome [15] available at the following website http://www.biostat.jhsph.edu/~hji/cisgenome.

To remove blob-like defects in array images, Song et al. developed Microarray Blob Remover (MBR) [35] (Fig. 12.2). This tool uses a two step algorithm to

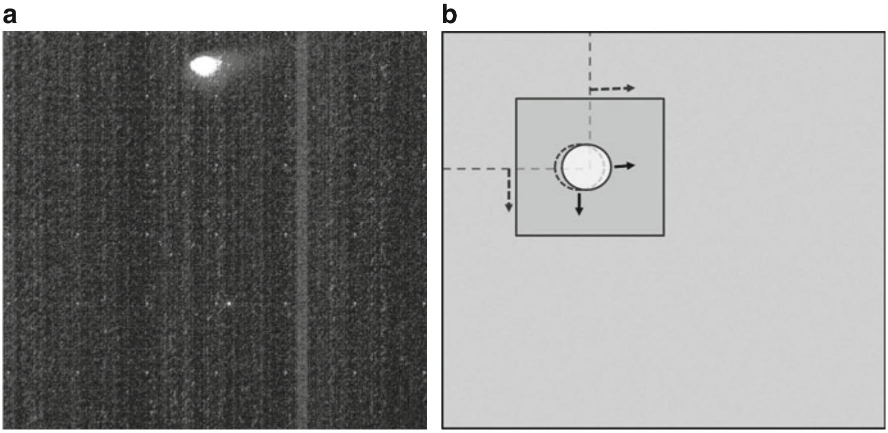


Fig. 12.2 Image artifacts. (**a**) The white area on the top shows an example of blob-like defects in array images. (**b**) MBR detects blob-like defects by first using a square to perform a coarse scan across the image and then using a circle to perform scan at a finer scale

automatically detect spatially clustered probes with high intensities. In the first step of the algorithm, a sliding square that covers 100 × 100 probes is used to scan the image with a step size of 50 probes. If more than half of the probes in the square have intensities exceeding a cutoff, the square will be labeled and will be examined further in the next step. The cutoff is chosen to be the kth percentile (default $k = 90$) of all probe intensities on the array. In the second step, squares labeled in the first step are analyzed at a finer scale. A circle of radius 20 is used to scan the squares with a step size of two probes. If more than p percent (default $p = 90$) of the probes in the circle have intensities exceeding the $(k - 5)$th percentile of overall probe intensities, then all probes in the circle will be labeled as outliers. The original data files (i.e. CEL files) will be updated to record the outlier information. Downstream analysis tools can then choose to exclude these outliers from analyses. MBR can be used to remove blob defects that occupy less than 10% of the array area. It has been shown that removing these artifacts improves detection of protein-DNA interactions. If the artifacts cover more than 10% of the probes, the authors of MBR suggest replacing the array by a new hybridization.

After removing image artifacts, the next step of preprocessing is to remove other systematic biases from the array data. There are two major types of biases. First, distributions of probe intensities generally vary across arrays. Some arrays are brighter than the others and have higher overall intensities. This is illustrated by boxplots in Fig. 12.3a which shows the probe intensity distributions of several arrays from a single ChIP-chip study. The difference could be attributed to a number of factors, including differences in scanner settings, amounts of reagents, room temperatures, and technician's experience levels, etc. This represents a bias that needs to be removed before meaningful comparisons across samples can be made. A procedure that removes this bias by matching the distribution of probe intensities across different array samples is called a normalization procedure.

The second type of bias is probe specific bias. Fig. 12.4a provides an example to illustrate this bias. The top six tracks in the figure show log2 probe intensities of three ChIP samples and three control samples in a typical ChIP-chip experiment. Probe intensities across different arrays have been normalized using quantile

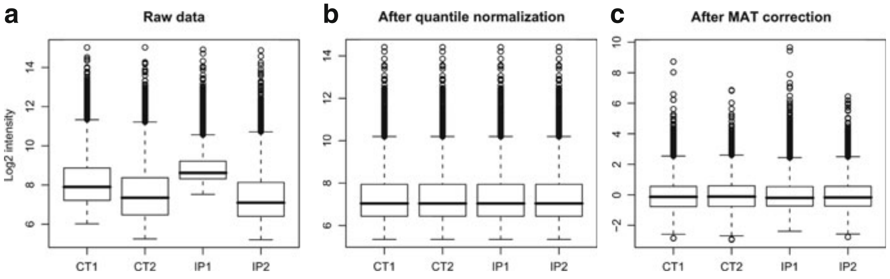


Fig. 12.3 Data normalization. (**a**) Distributions of raw log2 probe intensities of four different array samples. (**b**) Distributions after quantile normalization. (**c**) Distributions after MAT background correction

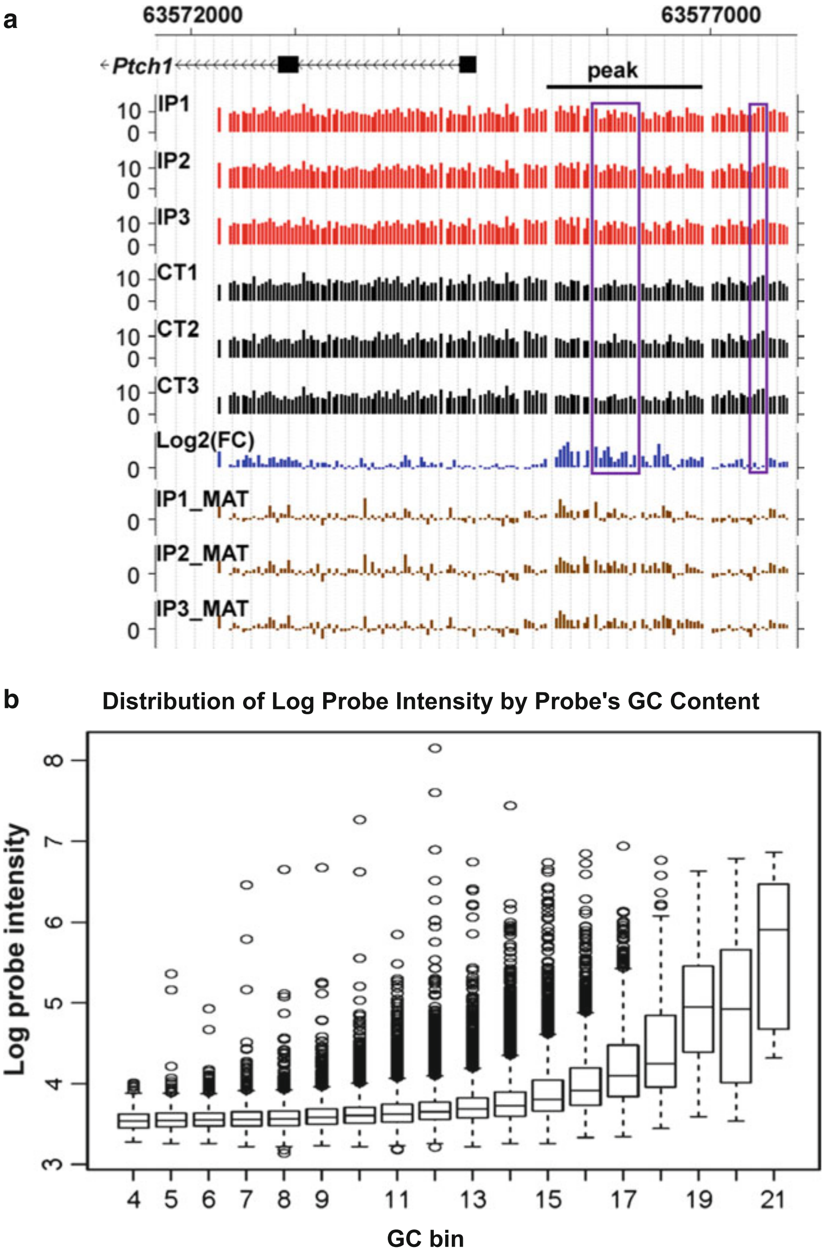


Fig. 12.4 Probe effects. (**a**) A ChIP-chip study for locating transcription factor binding sites of GLI3 protein. Affymetrix Mouse Promoter 1.0 R arrays were used. IP1-IP3, CT1-CT3: quantile normalized ChIP and control probe intensities at log2 scale. Log2(FC): log2(IP/CT) fold change. IP1_MAT-IP3_MAT: MAT background corrected probe intensities for IP1-IP3. (**b**) Probes are grouped into bins by GC content. Log probe intensities (logPM) from a typical array sample are shown for each bin. The plot shows that mean and variance of probe intensities are probe sequence dependent. Plot (**a**) is reproduced from [21]. Plot (**b**) is kindly provided by X. Shirley Liu

normalization (which will be introduced in Sect. 12.3.1). A biologically verified transcription factor binding site is indicated by the peak shown in the log2(FC) track which displays average log2 fold changes between the ChIP and control intensities. The figure clearly suggests that different probes tend to exibihit different intensity levels. Many probes outside the binding region have higher intensity values than probes inside the binding region (e.g. compare probes highlighted by the boxes). The trend is indeed consistent across different data sets [21]. These probe-specific behaviors, also known as probe effects [12, 20, 24, 38], are often dependent on probe sequences. To illustrate this, Fig. 12.4b grouped probes into bins based on their GC content. Probes with similar GC content were assigned to the same bin. Log probe intensities for each bin are shown for a typical array sample. The figure shows that GC-rich probes tend to have bigger intensities and bigger variability than AT-rich probes. As a result, GC-rich probes are more likely to show big fold changes in a comparison between two random samples. Removing this bias by appropriately modeling the probe effects can increase sensitivity and specifity of subsequent signal detection. A procedure that models and removes the probe effects is called a background correction procedure. A couple of methods have been proposed for normalization and background correction. Section 12.3 will provide a detailed discussion on this topic.

The final step of data preprocessing involves mapping the probes back to the genome. Each probe has a X-Y coordinate representing its physical location on the microarray. However, this coordinate does not tell you where in the genome this probe comes from. In order to know where protein-DNA interactions occur in the genome, one has to map the array coordinates to the genomic coordinates. Most array vendors provide this map. For example, the BPMAP files provided by Affymetrix contain information on genomic locations that each probe aligns to. Most signal detection tools use this information to convert X-Y coordinates in CEL files into genomic coordinates. Sometimes, the vendor provided map is based on an old genome assembly. If one wishes to perform analysis on the newest version of the genome, one may realign all probes on the array to the new assembly using tools such as xMAN [25] or SeqMap [18].

12.3 Background Correction and Normalization

Quantile normalization [6], MAT (Model based Analysis of Tiling arrays) [20] and TileProbe [21] are three major approaches for tiling array normalization and background correction. Quantile normalization attempts to normalize probe intensities across multiple arrays, whereas MAT and TileProbe are two background correction procedures that can be applied to detect signals even when a study does not contain control samples.

12.3.1 Quantile Normalization

Quantile normalization was initially designed for processing gene expression arrays. Now it has also been widely used in tiling array analysis. This method uses a simple transform to force different array samples to have a common probe intensity distribution. Let x_{ij} denote the raw intensity of probe i in array sample j. The quantile normalization first sorts probe intensities x_{ij} and arrange them from the smallest to the biggest within each sample j. Let $x_{(k)j}$ denote the k^{th} quantile of the probe intensities in sample j. The method computes the average of the k^{th} quantiles across samples, $y_{(k)} = \sum_{j=1}^{J} x_{(k)j}/J$, where J is the total number of samples. Then for each j, the probe intensity that corresponds to $x_{(k)j}$ is replaced by $y_{(k)}$. For example, if probe i in sample j has the intensity $x_{ij} = x_{(k)j}$, then x_{ij} is replaced by $y_{(k)}$. If the same probe has an intensity $x_{ij'} = x_{(k')j'}$ in sample j', then $x_{ij'}$ is replaced by $y_{(k')}$. After this transform, probe intensity distributions of all array samples become the same. Figure 12.3b shows the quantile normalized probe intensities for arrays in Fig. 12.3a. It clearly illustrates that after normalization, the systematic bias associated with different samples are removed.

An implicit assumption used by quantile normalization is that the majority of probes correspond to DNA species that do not change across samples. This assumption is reasonable in most ChIP-chip studies. This approach does not try to remove probe effects. Therefore, protein-DNA interactions can only be detected by comparing ChIP and control samples, through which probe specific behaviors can be properly controlled. If there were no control samples, looking at the quantile normalized ChIP intensities alone would incorrectly define locations of transcription factor binding sites, as shown by the first seven tracks in Fig. 12.4a.

12.3.2 MAT

It is known that the probe effects are closely related to probes' thermodynamic properties which are probe sequence dependent. Using this fact, the MAT method attempts to explain background probe intensities using probe sequences. It is assumed that most probes on an array measure background noise, which is reasonable in most ChIP-chip studies. With this assumption, MAT uses millions of probes on an array to fit a regression:

$$log(PM_i) = \alpha n_{iT} + \sum_{j=1}^{25} \sum_{k \in \{A,C,G\}} \beta_{jk} I_{ijk} + \sum_{k \in \{A,C,G,T\}} \gamma_k n_{ik}^2 + \delta log(c_i) + \epsilon_i \tag{12.1}$$

Here, PM_i represents intensity of a perfect match probe i; n_{ik} is the number of nucleotide k in probe i; I_{ijk} indicates whether the jth nucleotide of probe i is k ($I_{ijk} = 1$) or not ($I_{ijk} = 0$); c_i is the number of times the sequence of probe i occurs in the genome; α, β_{jk}, γ_k and δ are regression coefficients; and ϵ_i is the probe specific error. The model contains 81 regression coefficients. With millions of data points available, these coefficients can be robustly determined.

The regression is fit to each individual sample. Using the fitted parameters, log probe intensity of probe i can be predicted. Let $\hat{m}_i$ denote the predicted intensity of probe i. Probes are grouped into affinity bins based on $\hat{m}_i$, each containing approximately 3,000 probes with similar $\hat{m}_i$ values. Let s_i be the standard deviation of the affinity bin containing probe i. The MAT corrected probe intensities are then defined as:

$$t_i = \frac{log(PM_i) - \hat{m}_i}{s_i} \tag{12.2}$$

This procedure removes a significant fraction of sequence dependent probe behaviors. Figure 12.3c shows that for samples used in Fig. 12.3a, the distribution of MAT corrected probe intensities t_i have similar empirical distributions. In fact, the distributions are similar enough so that further normalization across samples is not needed [20].

MAT can be applied to individual samples. A single ChIP sample suffices the MAT analysis. The last three tracks of Fig. 12.4a show that after removing sequence dependent background by MAT, protein-DNA interaction signals can be detected even without control samples. This makes MAT a very attractive tool in pilot studies for which the main purpose is to test antibodies, or in studies that involve profiling a large number of biological samples in different cell types. In both scenarios, it is desirable to keep the cost low such as by using fewer samples.

12.3.3 TileProbe

The MAT model can remove a significant fraction of probe effects, however, it does not remove all probe effects. The top four tracks of Fig. 12.5 show MAT background corrected probe intensities for two ChIP and two control samples. Existence of residual probe effects is obvious in that a continuous run of probes show positive MAT corrected probe intensities not only in the ChIP samples but also in the control samples. The track named “MedianMAT_All-GEO-Arrays” in the same figure displays the median MAT corrected probe intensities of all array samples stored in the GEO database [4], which shows that the residual probe effects are consistent across different studies. Existence of residual probe effects could be explained by several factors. First, MAT uses an unsaturated model that includes only the main effects of probe sequence and a few squared terms as covariates. As a result, it cannot explain probe effects due to higher order interactions between nucleotides at different positions within a probe. Second, not all probe effects are sequence dependent (e.g. the physical location of a probe in the array may also contribute to the probe effects). As a result, the prediction of probe effects based on probe sequences may not be perfect.

The residual probe effects in the MAT corrected probe intensities could directly affect the subsequent detection of biological signals. For example, in Fig. 12.5, if one only had the ChIP samples in the first two tracks, MAT would report a high

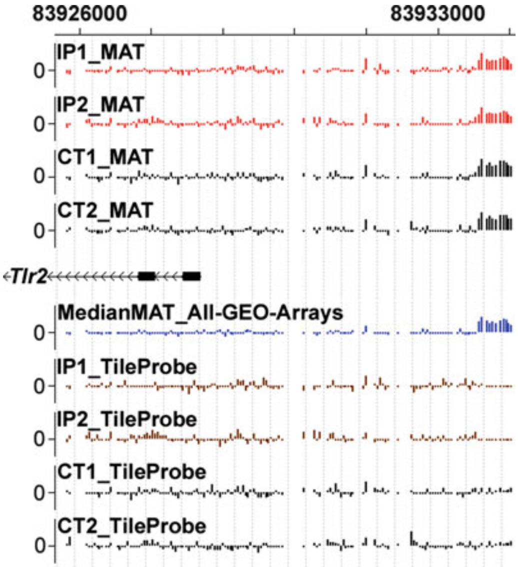


Fig. 12.5 Residual probe effects after MAT correction. Data from a ChIP-chip study using Affymetrix Mouse Promoter 1.0 R arrays are shown. IP_MAT, CT_MAT: MAT corrected probe intensities for ChIP and control samples respectively. MedianMAT_All-GEO-Arrays: median MAT corrected probe intensities across hundreds of samples stored in GEO database. IP_TileProbe, CT_TileProbe: TileProbe background corrected probe intensities

confidence peak. However, the comparison with control samples clearly illustrates that this region is a false positive. This example suggests that by removing the residual probe effects, one should be able to further improve the sensitivity and specificity of subsequent analysis.

Based on this observation, Judy and Ji developed another approach, TileProbe [21], to model probe effects. TileProbe takes advantage of the diverse and large number of samples stored in the GEO database and uses these publicly available data to obtain a robust model for MAT residual probe effects. To build the probe effect model for a particular array platform, TileProbe first applies MAT to each individual array sample collected from the GEO database. This database contains more than a hundred samples per platform for the commonly used array platforms. After this step, a MAT corrected intensity is attached to each probe for each sample. Next, all array samples are grouped according to studies and experimental conditions. For example, if a study (determined by the GEO series number) contains three ChIP samples and three control samples, the six samples will be divided into two groups: an IP group and a control group. Assume that there are G groups in total and group g ($g \in \{1, 2, \ldots, G\}$) contains K_g replicate samples. Let t_{igk} denote the MAT corrected probe intensity of probe i in the k^{th} replicate of group g, and $\bar{t}_{ig} = \sum_k t_{igk}/K_g$. TileProbe models the residual probe effects in t_{igk} using two quantities θ_i and τ_i which are determined as follows:

$$\theta_i = median\left\{t_{igk},\ g \in \{1, 2, \ldots, G\}\ and\ k \in \left\{1, 2, \ldots K_g\right\}\right\} \tag{12.3}$$

$$\omega_i^2 = \frac{\sum_{g=1}^{G} \sum_{k=1}^{K_g} (t_{igk} - \bar{t}_{ig})^2}{\sum_{g=1}^{G} (K_g - 1)} \tag{12.4}$$

$$\tau_i^2 = (1 - B)\omega_i^2 + B\overline{\omega^2} \tag{12.5}$$

Here $\overline{\omega^2}$ is the mean of all ω_i^2, and $B \in [0, 1]$ is a shrinkage factor computed using the variance shrinkage estimator in formula (12.13) which will be introduced in Sect. 12.4.2.2. In other words, TileProbe uses θ_i, the median MAT corrected probe intensity across all samples, to model the magnitude of each residual probe effect. This assumes that, at each probe, most samples used for building the probe model do not contain biologically relevant signals. The assumption holds when a large number of diverse samples, representing different experimental systems (e.g. different transcription factors in ChIP-chip experiments) and different conditions, are used for building the model. In addition, the probe specific variability is modeled by τ_i. The shrinkage estimator in formula (12.5) is used to avoid unstable variance estimates when the available degrees of freedom $\sum_g (K_g - 1)$ are small.

Using Eqs. 12.3–12.5, a probe effect model can be built for each array platform. When a new data set generated by the same platform needs to be analyzed, one can first apply MAT correction (i.e. formulas 12.1 and 12.2) to each sample, u. Next, the MAT corrected probe intensity, t_{iu} for probe i and sample u, is standardized as follows:

$$y_{iu} = \frac{t_{iu} - \theta_i}{\tau_i} \tag{12.6}$$

The y_{iu} statistic is the TileProbe background corrected probe intensity, which can be used as input for subsequent peak detection. If there is good reason to believe that the estimate of τ_i is not stable, a simplified version of TileProbe may be used in which $y_{iu} = t_{iu} - \theta_i$. In the following sections, this simplified version is denoted as TPM, and the original version with variance standardization (i.e. formula 12.6) is denoted as TPV.

Applying TileProbe solves the issues caused by residual probe effects. The bottom four tracks in Fig. 12.5 show TileProbe corrected probe intensities. Compared to the MAT corrected intensities, the residual probe effects no longer exist. Using TileProbe does not require availability of control samples.

12.3.4 Comparison of Normalization and Background Correction Methods

Quantile normalization, MAT and TileProbe were compared in a recent study [21]. The study analyzed four different ChIP-chip data sets stored in the GEO database. These data represent four transcription factors and two different array platforms (Affymetrix Mouse Promoter 1.0 R and Affymetrix Human Tiling 2.0R array 6). The data were first processed using the three methods described above, then a

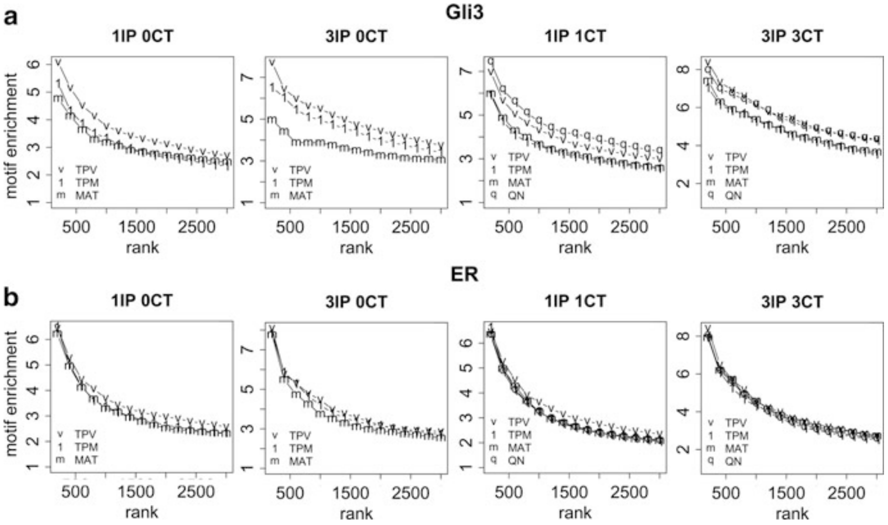


Fig. 12.6 Comparisons of quantile normalization, MAT and TileProbe. After peak detection, enrichment ratios of the relevant binding motifs among the top 200, 400, …, etc. binding regions are shown. The ratio was determined by comparing the percentage of ChIP-chip peaks that contained at least one motif site to the percentage of negative control peaks that contained $\geq$ 1 motif site. TPV: TileProbe-TPV; TPM: TileProbe-TPM; QN: quantile normalization

common peak detection protocol (i.e. the MAT peak calling algorithm that will be introduced in Sect. 12.4.2.3) was applied to detect TF binding regions. After ranked lists of binding regions were reported, enrichment of transcription factor binding motifs in the predicted binding regions were compared. Figure 12.6 displays the results for two data sets (Gli3 and ER). Each data set was analyzed under four different analytical conditions: 1IP 0CT (i.e. using one ChIP sample and no control sample), 1IP 1CT, 3IP 0CT, and 3IP 3CT. The results indicate that TileProbe outperformed MAT when there were no control samples. Without control samples, quantile normalization was not applicable. When control samples were available, all three algorithms performed similarly, but quantile normalization and TileProbe slightly outperformed MAT.

12.4 Signal Detection

After preprocessing, probes are sorted based on their genomic coordinates, and systematic bias is removed from probe intensities. We are now ready to detect locations of protein-DNA interactions. Since the DNA fragments in ChIP samples are longer than probe spacing, each fragment can cover multiple probes. As a result, a *bona fide* protein-DNA interaction is typically indicated by a continuous run of probes that show increased intensities in ChIP samples compared to the background noise (see e.g. Fig. 12.4). When designing signal detection algorithms, this spatial correlation could be used to improve the methods' discriminating power. A good method should also take into account that a ChIP-chip experiment can produce tens of millions of data points. For example, a study involving three ChIP and three control samples on Affymetrix human genome tiling arrays contains $\geq$250 million data points. Practically, it is important to be able to process the huge data sets within reasonable time. Finally, ChIP-chip experiments are expensive. Most studies generate only a small number of biological replicates ($\leq$3). This poses a challenge on estimating biological variability. Efficiently using information in the small replicate studies is therefore important. Up to now, many methods have been developed for locating signals in ChIP-chip data. This section reviews four major classes of them, including a method based on non-parametric test, moving average methods, methods based on Hidden Markov Models (HMM), and methods that use peak shape or kernel deconvolution.

12.4.1 Wilcoxon Rank-Sum Test

The Affymetrix Tiling Array Analysis Software (TAS) uses a non-parametric approach to detect signals [22]. This method first defines a bandwidth B. For each probe, a local data set is formed by collecting all probe intensities (from both ChIP and control samples) within $\pm B$ base pairs. Probe intensities (PM or PM-MM) within the local window are sorted and a rank-sum test is performed. The p-value obtained from the test is attached to the probe in question. After applying this procedure to all probes, probes with p-values smaller than a user-chosen cutoff are marked as positive probes. Positive probes are used to construct protein-DNA binding regions by merging neighboring probes into a single region if their distance $\leq$*max_gap* base pairs. Regions shorter than *min_run* base pairs are excluded, and the remaining regions are reported as protein-DNA interactions. Compared to the other methods, this non-parametric approach is not the most powerful one. However, it does not require parametric assumptions used by the other methods and is robust to deviations from those assumptions. This method requires a sorting operation for each window, which is time-consuming when a large data set is analyzed.

12.4.2 Moving Average Methods

Moving average is one of the most commonly used methods in ChIP-chip data analysis. Many popular software tools are based on moving average.

12.4.2.1 Average *T*-statistics

Keles et al. [23] uses a simple moving average based on *t*-statistics. Assume that there are I probes in the tiling array, J different types of DNA samples, and K_j

replicates for the jth type of sample. In most ChIP-chip studies, $J = 2$ (corresponding to ChIP and control); $j = 1$ denotes the ChIP sample, and $j = 2$ denotes the control sample. Let X_{ijk} denote the normalized (or background corrected) and log-transformed intensity of probe i in the kth replicate of sample j. Assume that probes are indexed according to their genomic coordinates (i.e. i and $i+1$ are two neighboring probes in the genome). $\overline{X}_{ij} = \sum_k X_{ijk}/K_j$, and $s_{ij}^2 = \sum_k (X_{ijk} - \overline{X}_{ij})^2/(K_j - 1)$. The method in [23] first computes a t-statistic for each probe:

$$t_i = \frac{\overline{X}_{i1} - \overline{X}_{i2}}{\sqrt{s_{i1}^2/K_1 + s_{i2}^2/K_2}} \tag{12.7}$$

For each probe i, it then collects $2W$ flanking probes (W on the left and W on the right) and computes a moving average statistic using $2W + 1$ t-statistics.

$$m_i = \frac{\sum_{k=i-W}^{i+W} t_k}{2W + 1} \tag{12.8}$$

Next, probes with m_i bigger than a cutoff are used to define protein-DNA interactions, and a procedure similar to TAS (see Sect. 12.4.1) is used to construct regions to be reported.

To choose an appropriate W, a cross-validation procedure was proposed. In order to control type I errors, m_i are converted to p-values. A nested-Bonferroni procedure was developed to control family wise error rate. This procedure considers correlation among the m_i statistics and is less conservative than Bonferroni adjustment. Readers are referred to [23] for a detailed description.

12.4.2.2 TileMap Moving Average

Most ChIP-chip experiments have a limited number of replicates. With small degrees of freedom, variance estimates are unstable. There are millions of probes. Just by chance, some of them have small sample variances (i.e. $s_{ij}^2 \approx 0$). These probes tend to have big t_i values, however they do not represent real biological signals. This is a major source of noise when applying the method described in Sect. 12.4.2.1 to make signal calls. TileMap moving average solves this problem by using a technique called “variance shrinking”. This method employs a hierarchical model to describe the data. The model allows one to pool information from all probes on the array to estimate the variance associated with individual probes. The same technique has been used in analyzing differentially expressed genes in microarray studies [3, 10, 34].

In TileMap, it is assumed that

$$X_{ijk} | \mu_{ij}, \sigma_i^2 \overset{ind.}{\sim} N(\mu_{ij}, \sigma_i^2) \tag{12.9}$$

$$\mu_{ij} \overset{ind.}{\propto} 1 \tag{12.10}$$

$$\sigma_i^2|\nu_0, \omega_0^2 \overset{i.i.d.}{\sim} Inv - \chi^2(\nu_0, \omega_0^2) \tag{12.11}$$

Define $\nu = \sum_j (K_j - 1)$, $s_i^2 = \sum_j \sum_k (X_{ijk} - \overline{X}_{ij})^2/\nu$, $\overline{s^2} = \sum_i s_i^2/I$ and $S = \sum_i [s_i^2 - \overline{s^2}]^2$. TileMap first estimates σ_i^2 using a closed-form empirical Bayes shrinkage estimator:

$$\hat{\sigma}_i^2 = (1 - \hat{B})s_i^2 + \hat{B}\overline{s^2} \tag{12.12}$$

where $\hat{B}$ is an estimator for $var(s_i^2|\sigma_i^2)/var(s_i^2)$ and is computed using

$$\hat{B} = \min(1, \frac{2}{\nu+2}\frac{I-1}{I} + \frac{2}{\nu+2}(\overline{s^2})^2\frac{I-1}{S}) \tag{12.13}$$

The shrinkage estimator pools information from all s_i^2 to estimate σ_i^2. It introduces additional degrees of freedom to variance estimates. Small s_{ij}^2 are pulled away from zero. The variance estimates are more stable and can have a smaller ensemble mean square error when all σ_i^2 are considered jointly.

Once $\hat{\sigma}_i^2$ is obtained, TileMap replaces Eq. 12.7 by:

$$t_i = \frac{\overline{X}_{i1} - \overline{X}_{i2}}{\sqrt{(1/K_1 + 1/K_2)\hat{\sigma}_i^2}} \tag{12.14}$$

Formula (12.14) is then plugged into formula (12.8) to compute the moving average statistics, which will be used to find protein-DNA interactions.

Variance shrinking greatly improves the statistical power of signal detection. Figure 12.7a illustrates how this method works in a ChIP-chip study involving transcription factor GLI1. When log2 ratios between ChIP and control probe intensities were plotted, there were no clear peaks. Using the *t*-statistics in formula (12.7), some weak signals emerged. However, given the high level of noise, it remains unclear whether they are real peaks or not. When formulas (12.12–12.14) were used to compute the *t*-statistics with variance shrinking, three peaks became clear, and the signals were further improved after taking the moving average. The peak indicated by the arrow was actually tested in [36] using transgenic experiments and was verified to be a functional *cis*-regulatory element.

Figure 12.7b compares TileMap moving average with the moving average without variance shrinking. A cMyc dataset with 6 ChIP samples and 12 control samples was analyzed. First, all 18 samples were analyzed using the non-shrinking method, and the regions reported were treated as gold standard. Next, a subset of data containing only two ChIP and two control samples were selected to serve as test data. The non-shrinking method was applied to analyze the reduced data set. The test data were further reduced by excluding half of the probes, and the shrinking method was applied to this further reduced data set to detect signals. The shrinking and non-shrinking methods were then compared in terms of how many gold standard regions they found among the top predictions. The evaluation was repeated by choosing

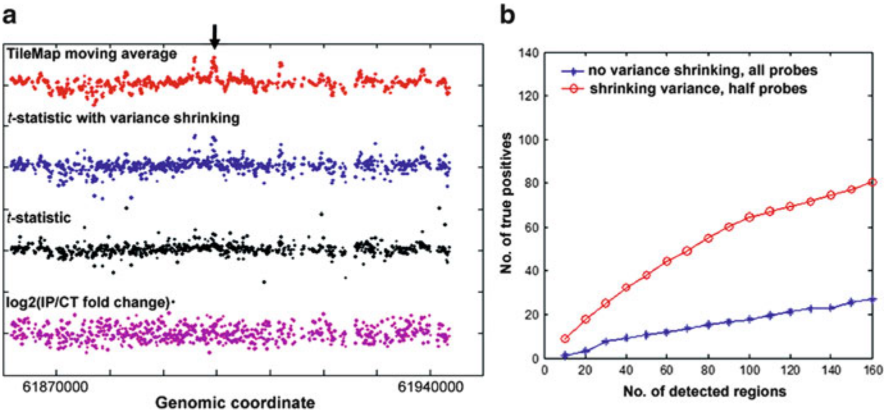


Fig. 12.7 Variance shrinking improves signal detection. (**a**) Analysis of a ChIP-chip data set for transcription factor GLI1. Each dot is a probe. From *bottom to top*, the four tracks are log2 fold change between IP and control, *t*-statistics without variance shrinking, *t*-statistics with variance shrinking, and TileMap moving average. (**b**) Comparison of shrinking and non-shrinking methods using a cMyc data set. The figure shows the number of gold standard regions reported by the two methods among their top 10, 20, ..., etc. predictions

different combinations of array samples to form the test data. Figure 12.7b shows the average performance. Although the gold standard was constructed in favor of non-shrinking method and the non-shrinking method used twice as many probes, the shrinking method outperformed the non-shrinking one significantly.

TileMap moving average is computationally efficient since it does not require sorting data repeatedly as TAS. In order to control false discovery rate (FDR), an unbalanced mixture subtraction (UMS) method was proposed. This method controls FDR at the probe level and is empirically very conservative. Other options for estimating FDR, including a permutation method and a method that estimates FDR using the left tail of the empirical distribution of m_i, are also provided. The left-tail method, which will be introduced in the next section, estimates FDR at region level. In other words, instead of reporting what percentage of probes in the reported regions are expected to be false, it estimates what percentage of reported regions are expected to be false. Details of these methods are presented in [15, 16].

12.4.2.3 MAT

MAT uses a robust version of moving average as its peak calling algorithm. It uses a W bp sliding window to scan the genome. The default value of W is 600 bp.

If a study contains ChIP sample(s) only, MAT computes a MATscore for each window. The MATscore is defined as follows:

$$MATscore(window\ k) = \sqrt{n_k} \times TM(t_i\ in\ window\ k) \tag{12.15}$$

Here t_i is the MAT background corrected intensities determined by Eq. 12.2; $TM(.)$ is the trimmed mean of all t_i within the window; n_k is the number of data points in the window. The trimmed mean removes the top 10% and bottom 10% of t_i values. If a window contains $< l$ probes (usually $l = 10$), then the window is excluded from the analysis. By using the trimmed mean, MAT is robust to isolated outlier probes, i.e. probes that have high t_i values but are surrounded by probes with no enrichment signals.

If control samples are available, the MATscore is replaced by:

$$MATscore(window\ k) = \sqrt{n_{k,ChIP}} \times \frac{TM(t_i\ from\ ChIP\ samples) - TM(t_i\ from\ control\ samples)}{\sigma_{control}} \tag{12.16}$$

Here, $n_{k,ChIP}$ is the number of data points from the ChIP samples in the window. $\sigma_{control}$ is the standard deviation of t_i in control samples. The MAT authors recommend that $\sigma_{control}$ only be used when there are at least three replicate control samples. If this condition is not satisfied, it is recommended that the difference between the ChIP and control TM values should be used instead [20].

Once MATscores are computed, one can collect all probes for which MATscores are bigger than certain cutoff. These probes are used to construct protein-DNA binding regions. To determine the false discovery rate, it is assumed that the MATscore follows a symmetric distribution centered at zero when no protein-DNA interactions exist. Under this assumption, one can flip the sign of MATscores and repeat the signal detection procedure using the same cutoff. All regions reported in the second analysis are false positives. They provide an estimate of the expected number of false discoveries. The ratio (No. of expected false positives / No. of detected protein-DNA interactions) then provides an estimate of FDR. MAT requires one to sort data within local windows in order to compute the trimmed mean, therefore it requires more computation than TileMap moving average.

12.4.3 Hidden Markov Models

Hidden Markov Model (HMM) is another popular method in ChIP-chip analysis. In the simplest two-state HMM, it is assumed that each probe has two possible states: a non-enriched (or background) state represented by 0, and an enriched (or protein-DNA interaction) state represented by 1. For probe i, the intensities associated with state 0 and 1 are governed by emission probability distributions $f_{i0}(x)$ and $f_{i1}(x)$ respectively. The transition between state 0 and 1 is governed by a transition probability matrix. With these assumptions, once the model parameters are given, the standard forward-backward algorithm can be applied to compute the posterior probabilities that probes are in state 1. Probes with these posterior probabilities bigger than certain cutoff (e.g. 0.5) then define protein-DNA interactions. A nice introduction to HMM and the forward-backward algorithm can be found in [11].

In a tool called HMMTiling [26], the emission probability $f_{i0}(x)$ is assumed to be a normal distribution $N(\mu_i, \sigma_i^2)$. μ_i and σ_i^2 are estimated using control samples from previous ChIP-chip studies. $f_{i1}(x)$ is assumed to be $N(\mu_i + 2\sigma_i, (1.5\sigma_i)^2)$. The transition probability from 0 to 1 and from 1 to 0 is assumed to be J/K, where J is a prior estimate of the number of potential protein-DNA interactions, and K is the total number of probes.

TileMap also provides a HMM routine to detect protein-DNA interactions. Instead of modeling probe intensities directly, TileMap HMM first summarizes enrichment information at each probe using formula (12.14). It is assumed that under state 0 and 1, the emission probability for t_i is $f_0(t)$ and $f_1(t)$ respectively. $f_0(t)$ and $f_1(t)$ are estimated using the unbalanced mixture subtraction approach. Because TileMap HMM models the probe level summary statistics instead of probe intensities, it can be easily generalized to analyze a multiple condition experiment. To analyze such experiments, one only needs to replace the t-statistic in formula (12.14) by the posterior probability of a user-specified pattern computed under the hierarchical model (12.9–12.11).

In the two-state HMM described above, duration of the enriched state follows a geometric distribution a priori. This assumption is not ideal because the DNA fragment lengths in real ChIP samples have a distribution with a mode centered around 300–500 bp. Incorporating a more appropriate length distribution may help discriminate signals from noise better, which requires development of new model frameworks.

12.4.4 Peak Shape and Kernel Deconvolution

Signals in ChIP-chip data often have characteristic shapes. For example, the log2(IP/control) fold changes around transcription factor binding sites are usually triangle- or bell-shaped (see Fig. 12.7). The shape provides additional information for detecting signals from noise.

Zheng et al. [39] developed a Mpeak method that tries to use the peak shape information in the ChIP-chip analysis. Mpeak uses a Poisson point process to describe the procedure that generates the ChIP data. It is assumed that a genome sequence contain M potential binding sites $B_1, \ldots, B_M$. The probability that binding site B_m is bound by protein is p_m. When the sequence is sheared, the probability that a cut occurs between a small interval $(x, x + \Delta x)$ is $\lambda(x)\Delta x$. After cutting, the genome sequence becomes a set of non-overlapping fragments. The probability that a protein-bound site is immunoprecipitated by antibody is α. For simplicity, one can first consider data that only contain one binding site B_m. Assume that its coordinate is 0. Sufficient and necessary conditions for a probe at $x > 0$ to be covered by a ChIP fragment is that the fragment covers both 0 and x (i.e. there is no cutting point between 0 and x), B_m is bound by the protein, and immunoprecipitated by the antibody. Based on the Poisson process assumption,

$$logPr(no\ cut \in (0,x)) \approx \sum_{i=1}^{n} log\left(1 - \lambda\left(\frac{ix}{n}\right)\frac{x}{n}\right) \overset{n\to\infty}{\to} -\int_0^x \lambda(s)ds \quad (12.17)$$

The probability that x is covered by a ChIP fragment, $p(x)$, therefore can be represented by $logp(x) = log(p_m\alpha) - \int_0^x \lambda(s)ds$ for $x > 0$. If $\lambda(x) = a$ for $x > 0$, then $logp(x) = c - ax$ where $c = log(p_m\alpha)$. Similarly for $x < 0$, if $\lambda(x) = b$, then $logp(x) = c + bx$. Together, this implies that $logp(x)$ has a triangle shape determined by:

$$log\,p(x) = c + bxI(x < 0) - axI(x \geq 0) \quad (12.18)$$

where $I(.)$ is an indicator function which is equal to one if its argument is true and equal to 0 otherwise. Each DNA sample contains many copies of genome sequences, and they all undergo a similar procedure. If it is assumed that probe intensities (after appropriate normalization and transformation) are proportional to $log\,p(x)$, then the model provides a probabilistic explanation of why the observed signals look like triangles. Real data contain more than one binding sites. For multiple binding site data, one can modify $p(x)$ slightly to reflect the joint effects of all binding sites (see [39] for details). If the binding sites are not close and do not interfere with each other, the triangle model can be used to describe the observed peak shape reasonably well. Based on this model, Mpeak attempts to fit triangle-shaped regressions to data around each probe. Locations of binding sites can then be determined according to the goodness of fit.

The Poisson point process used by Mpeak is an idealized model to describe data. In many data sets, peaks are not triangular. To deal with these data, Qi et al. [31] proposed the joint binding deconvolution (JBD) method. This approach relies on empirically determined DNA fragment length distribution which can be obtained using experimental techniques. Once this distribution is available, it is used to derive an influence function that describes the signal strength at various distances from a binding site. The influence function is able to capture shapes other than triangles. It is used as a kernel to deconvolve the observed data. Locations and strength of binding sites are inferred based on the deconvolution results. Using this approach, one can identify binding sites at a resolution higher than the probe spacing.

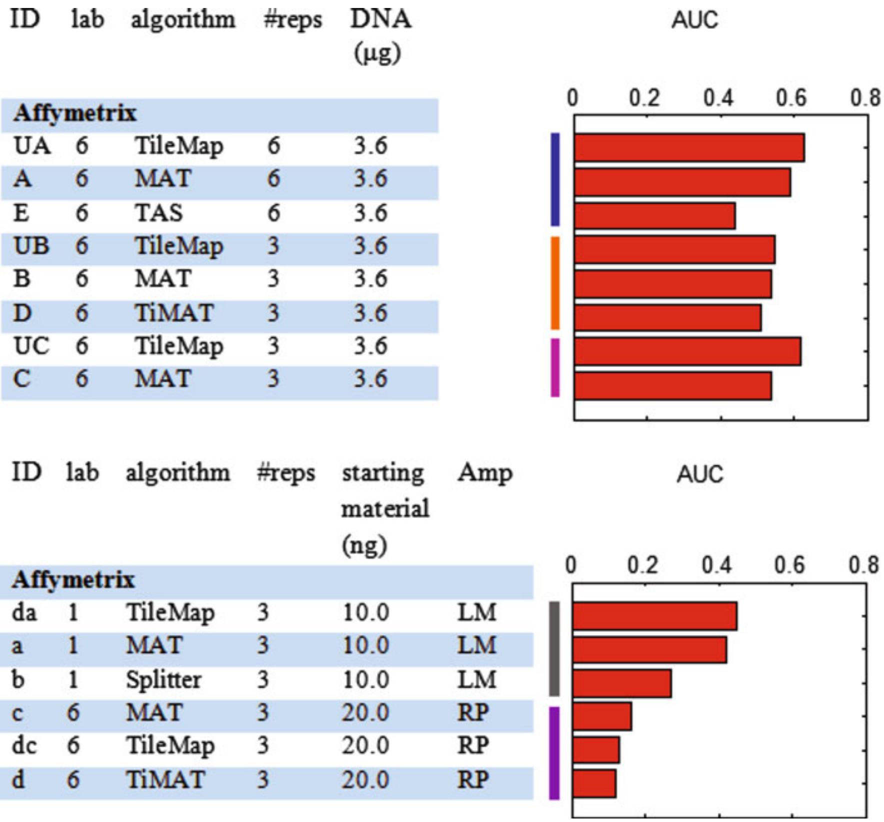
Mpeak and JBD were originally designed for processing NimbleGen and Agilent arrays. Although they are not directly designed for Affymetrix array data, the ideas behind them are general and should be applicable to all array platforms.

12.4.5 Methods Evaluation

Developing powerful signal detection algorithms for ChIP-chip data analysis is important. Equally important is to have an objective approach to evaluate different algorithms. For the purpose of evaluation, one needs to know where true protein-DNA interactions are, and one needs to have objective criteria to measure algorithms' performance. Having a comprehensive list of true protein-DNA interactions is difficult. To circumvent this difficulty, Johnson et al. [19] generated a spike-in data set for testing ChIP-chip signal detection algorithms. The spike-in samples were created by mixing 100 cloned promoter sequences (i.e. spike-ins, each about 500 bp) with human genomic DNA. The spike-in clones were added at different concentration levels. The samples were hybridized to Affymetrix, Agilent and NimbleGen ENCODE arrays by different labs. With about 100 true signals known, data generated by this study provide a good benchmark for evaluating ChIP-chip signal detection algorithms.

Using the spike-in data, Johnson et al. [19] compared the receiver operating characteristic (ROC) curve of a number of signal detection tools. Later, the results were supplemented by [15] adding the ROC curve of TileMap. Figure 12.8 shows the area under the ROC curve (AUC) for the compared algorithms designed for Affymetrix tiling array analysis. A bigger AUC value indicates better sensitivity and specificity. Based on this criterion, TileMap and MAT performed best. In addition to a ranked list of predictions, each tool also provides a cutoff that defines how many peaks should be reported. To assess the cutoffs chosen by different tools, [19] defined the optimal cutoff as the point on the ROC curve that is closest to the upper left corner (i.e. the point with coordinate (0,1)) of the sensitivity-specificity plot. The distance between the algorithm-chosen cutoff and the optimal cutoff is called E-O distance. A small E-O distance means that the cutoff chosen by the algorithm is neither too conservative nor too optimistic. When different tools were compared in terms of their E-O distances, MAT and TileMap again performed among the best. Similar comparisons of AUC and E-O distances were also performed for Agilent and NimbleGen arrays. Readers can find details and results in [19] and [15]. These studies did not test all available algorithms, therefore the available comparison results may not be comprehensive. However, the spike-in data and the performance criteria described above provide a mechanism to objectively evaluate other methods in future.

Evaluation based on the spike-in data has its own limitations. The current data were generated using ENCODE arrays that cover only a small fraction of the genome. In addition, the spike-ins do not capture all characteristics of real protein-DNA interactions. For example, instead of having a triangle shape, many spike-ins show plateau-like signals since they are created by cloning a whole segment of promoter sequences. As a result, algorithms that use a particular peak shape to detect signals (e.g. Mpeak) may not receive objective assessment. Fortunately, many other evaluation methods are available as substitutes of the spike-in data. Many transcription factors have known binding motifs. For ChIP-chip experiments involving these transcription factors, motif enrichment in the TF binding regions can be used to evaluate different algorithms (see e.g. Fig. 12.6). If gene expression data are available, different algorithms can also be compared in terms of what percentage of protein-DNA interactions are associated with changes of gene expression. There could be many other examples. The general idea is to find an independent source of information that can verify the plausibility of the predicted protein-DNA interactions.



ID	lab	algorithm	#reps	DNA (μg)
Affymetrix				
UA	6	TileMap	6	3.6
A	6	MAT	6	3.6
E	6	TAS	6	3.6
UB	6	TileMap	3	3.6
B	6	MAT	3	3.6
D	6	TiMAT	3	3.6
UC	6	TileMap	3	3.6
C	6	MAT	3	3.6

ID	lab	algorithm	#reps	starting material (ng)	Amp
Affymetrix					
da	1	TileMap	3	10.0	LM
a	1	MAT	3	10.0	LM
b	1	Splitter	3	10.0	LM
c	6	MAT	3	20.0	RP
dc	6	TileMap	3	20.0	RP
d	6	TiMAT	3	20.0	RP

Fig. 12.8 Areas under the ROC curve (AUC) of several signal detection tools for Affymetrix data analysis. AUC is assessed using the spike-in data in [19]. The figure shows results for five different data sets (indicated by five vertical bars). In the *top panel*, no amplification was applied to prepare the ChIP samples for hybridization. In the *bottom panel*, two different amplification protocols (LM: ligation-mediated PCR and RP: random-priming PCR) were used to prepare the samples. The lab number 1 and 6 indicate two different labs that generated the data. #reps indicates how many ChIP samples are used in the analysis (the number of control samples are the same). (The figure is reproduced from Supplementary Fig. 14 in [15] which was obtained by redrew Fig. 2 of [19] by adding TileMap results)

12.5 Motif Analysis

ChIP-chip enables us to locate transcription factor binding at resolution of 500–1,000 bp. The resolution is largely determined by the DNA fragment length in the ChIP sample. Sequence motifs recognized by transcription factors are generally much shorter. Most motifs contain only 6–30 nucleotides. With further computational analysis, these short motifs can be identified within the reported binding regions, and thereby dramatically increase the resolution of binding site prediction.

The predicted motif sites may serve as candidates for further experimental studies such as knock-out and transgenic experiments.

There are two types of motif analysis. If the motif recognized by the transcription factor of interest is known from previous studies, one only needs to map it to regions reported by ChIP-chip. Tools including CisGenome [15] and MAST [2] can be used to fulfill this task. For many transcription factors, the binding motifs are unknown. If this is the case, then the motif needs to be discovered from the binding data. Often, the motif can be found by searching for sequence patterns enriched in the TF binding regions. Many *de novo* motif discovery algorithms have been developed to handle this task [1, 28, 30]. Readers are referred to [13] for a detailed review on *de novo* motif discovery.

When applied to ChIP-chip data, a *de novo* motif search may return multiple motifs. Often, it is not clear which one is the key motif recognized by the transcription factor involved in the study. A simple way to discriminate the functionally relevant motifs from irrelevant ones is compare these motifs' enrichment levels. Ideally, by comparing the occurrence rates of the motifs (i.e. No. of motif sites/1 kb) in binding regions to those in negative control regions, the key motif should have the highest level of enrichment. Unfortunately, an analysis of multiple ChIP-chip data sets performed in [14] indicates that this is not true if the negative control regions are randomly chosen from the genome. In fact, binding regions identified by ChIP-chip and regions randomly chosen from the genome can have very different characteristics. For example, they may have different GC content, or the TF binding regions may be located closer to promoters. Differences in motif occurrence rate may reflect different nucleotide compositions of these two types of regions. To deal with this problem, Ji et al. [14] proposed to use "matched genomic controls" instead of random controls as the baseline to evaluate motif enrichment. "Matched genomic controls" are control regions that are carefully chosen to match the genomic distribution of ChIP-chip reported binding regions. To choose these control regions for a ChIP-chip data set, binding regions reported by signal detection algorithms are annotated with their closest genes. Distances between centers of TF binding regions and their neighboring genes' transcription start sites are computed. Next, genes are randomly selected from the gene annotation database. For each chosen gene, a genomic region with pre-specified length is picked up in a way such that the distance between the region center and the gene's TSS follows the same empirical distribution of distances between ChIP-chip binding regions and their closest genes.

Using this method, Ji et al. [14] analyzed ChIP-chip data for multiple transcription factors. With the matched controls, the key motifs were successfully identified as the top ranking motif in all test data. Figure 12.9 shows an example that involves transcription factor Oct4. In this data set, *de novo* motif discovery reported more than ten motifs, some of which had stronger signals than the Oct4 motif when ranked by the MDSCAN score proposed in [30]. When random genomic controls were used to compute motifs' enrichment levels, the Oct4 motif did not rank as the highest. However, when the matched controls were used, it was clearly the top one among all the motifs. This indicates that with the help of matched control regions, we are able to unambiguously identify motifs responsible for the transcription factor binding in ChIP-chip studies.

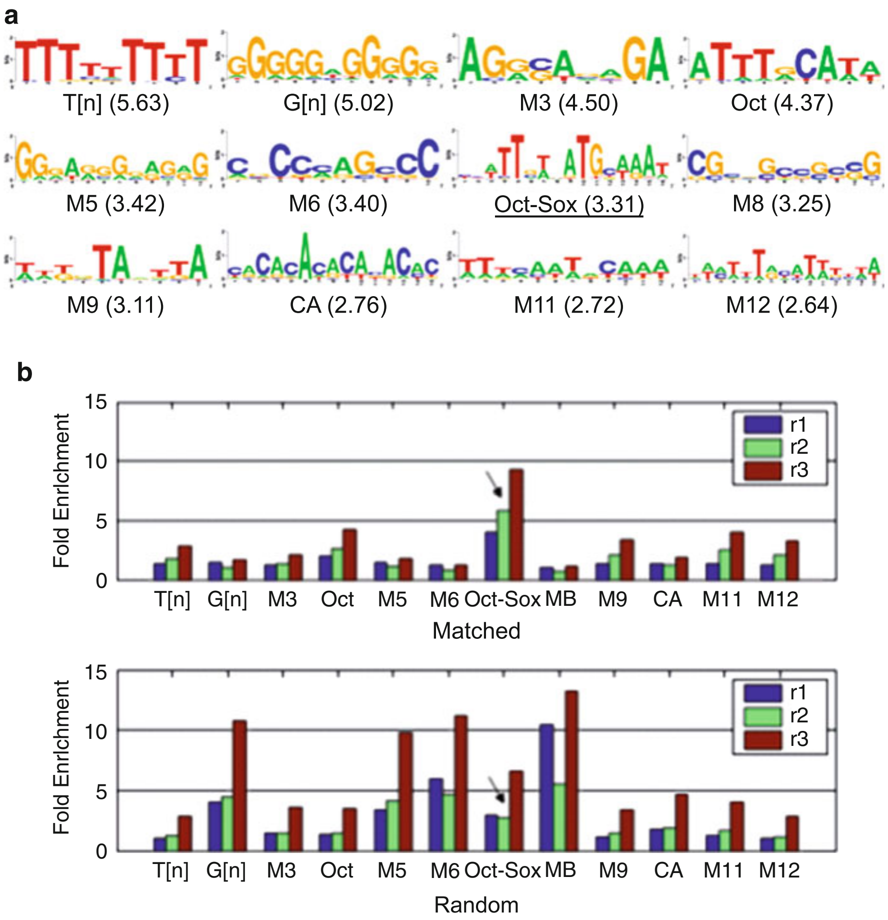


Fig. 12.9 Motif analysis of an Oct4 ChIP-chip study. (**a**) Motifs reported by a *de novo* discovery method, Gibbs motif sampler [28], are shown. For each motif, the MDSCAN score is shown in brackets. The score reflects both the motif's information content and the number of motif sites in the binding data [30]. The Oct4-Sox2 composite motif responsible for the sequence-specific protein binding is underlined. (**b**) For each motif, the relative enrichment levels r_1, r_2 and r_3 are shown as a group of three bars from left to right. Assume n_{1B} = no. of motif sites in TF binding regions, n_{2B} = no. of non-repeat base pairs in binding regions, n_{1C} = no. of motif sites in control regions, n_{2C} = no. of non-repeat base pairs in control regions, n_{3k} (k = B or C) = no. of phylogenetically conserved motif sites in specified genomic regions, and n_{4k} = no. of phylogenetically conserved non-repeat base pairs in the regions. $r_1 = (n_{1B}/n_{2B})/(n_{1C}/n_{2C})$; $r_2 = (n_{3B}/n_{4B})/(n_{3C}/n_{4C})$; $r_3 = (n_{3B}/n_{2B})/(n_{3C}/n_{2C})$. The relative enrichment levels are computed using either matched genomic controls ("*Matched*") or random genomic controls ("*Random*"). Relative enrichment levels of different motifs are compared, and the Oct4-Sox2 motif responsible for the binding is indicated by the arrows. (The figure is modified from [14])

12.6 Concluding Remarks

Data preprocessing, signal detection and motif analysis are three important steps of ChIP-chip data analysis. Methods discussed in this chapter represent a selected collection of available methods that have been widely used or that have potential to become so. They cover all three aspects described above. These methods can help one efficiently extract meaningful information from ChIP-chip data and allow us to locate protein-DNA interactions in the genome at a high resolution. Locating protein-DNA interactions represents the first step to use ChIP-chip data. A more important scientific question is what are the functions of the reported protein-DNA interactions. Recent studies in different biological systems suggest that a large proportion of transcription factor binding sites identified by ChIP-chip correspond to inert binding without playing roles in activating or repressing gene expression [27, 37]. This highlights the importance of further analyses that aim to characterize functions of ChIP-chip signals. To infer functions of protein-DNA interactions, ChIP-chip data need to be examined together with other sources of information including gene expression, gene ontology, epigenetic marks, etc. The possible combinations of analyses are huge. Development of statistical methods that systematically address this issue is still at its infancy.

With the rapid development of high throughput sequencing technologies, many applications of tiling arrays can now find their counterparts based on the next generation sequencing (see [33] for a review). The sequencing counterpart of ChIP-chip is ChIP-seq which can provide better resolution for locating transcription factor binding sites. In the near future, however, ChIP-chip will remain to be an important tool for various genome-wide studies due to its relatively low cost and relatively mature protocols. Indeed, new ChIP-chip data sets are flowing into the public databases every month. In this context, there is continuing need for gaining better understanding of the current data processing techniques and developing better methods for data analysis. More importantly, as the amount of data in the public databases increases, there is an increasing need to compare and jointly analyze multiple ChIP-chip and ChIP-seq data sets to reveal regulatory programs behind the tightly specified temporal and spatial gene expression patterns. Methods targeted for this purpose are greatly needed, and this is an area that will challenge statistician, computer scientists, as well as biologists in the next couple of years.

Last but not least, there is an urgent need to translate efficient statistical and computational algorithms to user friendly software tools. ChIP-chip analysis involves huge data sets and complex analysis pipelines. This makes data analysis a daunting task for bench biologists with little training in programming. On the other hand, more and more labs continue to generate ChIP-chip data. This creates a bottleneck for data analysis. Software tools that can make the analysis procedure easily accessible to biologists are very useful. The recently developed tools such as CisGenome [15] and CEAS [17] partially fulfill this goal. However, they are mainly designed for performing basic analyses such as those reviewed in this chapter. For more advanced analyses such as data integration, new tools need to be developed.

With the help of these tools and statistical methods behind them, we would expect to learn much more from the ChIP-chip studies.

Acknowledgements This work is partially supported by the Johns Hopkins Faculty Professional Development Fund to H.J. The author would like to thank Jennifer T. Judy for helpful comments and proofreading the draft of this chapter.

References

1. Bailey, T. L., & Elkan, C. (1994). Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proceedings of the second international conference on intelligent systems for molecular biology* (pp. 28–36). Menlo Park, California, USA: AAAI Press.
2. Bailey, T. L., & Gribskov, M. (1998). Combining evidence using p-values: Application to sequence homology searches. *Bioinformatics*, *14*, 48–54.
3. Baldi, P., & Long, A. D. (2001). A Bayesian framework for the analysis of microarray expression data: Regularized t-test and statistical inferences of gene changes. *Bioinformatics*, *17*, 509–519.
4. Barrett, T., Troup, D. B., Wilhite, S. E., et al. (2007). NCBI GEO: Mining tens of millions of expression profiles – database and tools update. *Nucleic Acids Research*, *35*(Database issue), D760–765.
5. Bernstein, B. E., Mikkelsen, T. S., Xie, X., et al. (2006). A bivalent chromatin structure marks key developmental genes in embryonic stem cells. *Cell*, *125*, 315–326.
6. Bolstad, B. M., Irizarry, R. A., Astrand, M., & Speed, T. P. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, *19*, 185–193.
7. Boyer, L. A., et al. (2005). Core transcriptional regulatory circuitry in human embryonic stem cells. *Cell*, *122*, 947–956.
8. Carroll, J. S., et al. (2006). Genome-wide analysis of estrogen receptor binding sites. *Nature Genetics*, *38*, 1289–1297.
9. Cawley, S., et al. (2004). Unbiased mapping of transcription factor binding sites along human chromosomes 21 and 22 points to widespread regulation of noncoding RNAs. *Cell*, *116*, 499–509.
10. Cui, X., Hwang, J. T. G., Qiu, J., et al. (2005). Improved statistical tests for differential gene expression by shrinking variance components estimates. *Biostatistics*, *6*, 59–75.
11. Durbin, R., Eddy, S. R., Krogh, A., & Mitchison, G. (1998). *Biological sequence analysis – probabilistic models of proteins and nucleic acids.* Cambridge: Cambridge University Press.
12. Irizarry, R. A., Hobbs, B., Collin, F., et al. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, *4*, 249–264.
13. Jensen, S. T., Liu, X. S., Zhou, Q., & Liu, J. S. (2004). Computational discovery of gene regulatory binding motifs: A Bayesian perspective. *Statistical Science*, *19*, 188–204.
14. Ji, H., Vokes, S. A., & Wong, W. H. (2006). A comparative analysis of genome-wide chromatin immunoprecipitation data for mammalian transcription factors. *Nucleic Acids Research*, *34*, e146.
15. Ji, H., Jiang, H., Ma, W., et al. (2008). An integrated software system for analyzing ChIP-chip and ChIP-seq data. *Nature Biotechnology*, *26*, 1293–1300.
16. Ji, H., & Wong, W. H. (2005). TileMap: Create chromosomal map of tiling array hybridizations. *Bioinformatics*, *21*, 3629–3636.
17. Ji, X., Li, W., Song, J., Wei, L., & Liu, X. S. (2006). CEAS: cis-regulatory element annotation system. *Nucleic Acids Research*, *34*, W551–554.

18. Jiang, H., & Wong, W. H. (2008). SeqMap: Mapping massive amount of oligonucleotides to the genome. *Bioinformatics*, *24*, 2395–2396.
19. Johnson D. S., et al. (2008). Systematic evaluation of variability in ChIP-chip experiments using predefined DNA targets. *Genome Research*, *18*, 393–403.
20. Johnson, W. E., Li, W., Meyer, C. A., et al. (2006). Model-based analysis of tiling-arrays for ChIP-chip. *Proceedings of the National Academy of Sciences of the United States of America*, *103*, 12457–12462.
21. Judy, J. T., & Ji, H. (2009). TileProbe: Modeling tiling array probe effects using publicly available data. *Bioinformatics*, *25*, 2369–2375.
22. Kampa, D., et al. (2004). Novel RNAs identified from an in-depth analysis of the transcriptome of human chromosomes 21 and 22. *Genome Research*, *14*, 331–342.
23. Keles, S., van der Laan, M. J., Dudoit, S., & Cawley, S. E. (2006). Multiple testing methods for ChIP-Chip high density oligonucleotide array data. *Journal of Computational Biology*, *13*, 579–613.
24. Li, C., & Wong, W. H. (2001). Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proceedings of the National Academy of Sciences of the United States of America*, *98*, 31–36.
25. Li, W., Carroll, J. S., Brown, M., & Liu, X. S. (2008). xMAN: Extreme MApping of OligoNucleotides. *BMC Genomics*, *9*(Suppl. 1), S20.
26. Li, W., Meyer, C. A., & Liu, X. S. (2005). A hidden markov model for analyzing ChIP-chip experiments on genome tiling arrays and its application to p53 binding se-quences. *Bioinformatics*, *21*(Suppl. 1), i274–i282.
27. Li, X. Y., MacArthur, S., & Bourgon, R. (2008). Transcription factors bind thousands of active and inactive regions in the Drosophila blastoderm. *PLoS Biology*, *6*, e27.
28. Liu, J. S., Neuwald, A. F., & Lawrence, C. E. (1995). Bayesian models for multiple local sequence alignment and Gibbs sampling strategies. *Journal of the American Statistical Association*, *90*, 1156–1170.
29. Liu, X. S. (2007). Getting started in tiling microarray analysis. *PLoS Computational Biology*, *3*, e183.
30. Liu, X. S., Brutlag, D. L., & Liu, J. S. (2002). An algorithm for finding protein-DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments. *Nature Biotechnology*, *20*, 835–839.
31. Qi, Y., et al. (2006). High-resolution computational models of genome binding events. *Nature Biotechnology*, *24*, 963–970.
32. Ren, B., Robert, F., Wyrick, J. J., et al. (2000). Genome-wide location and function of DNA binding proteins. *Science*, *290*, 2306–2309.
33. Shendure, J., & Ji, H. (2008). Next-generation DNA sequencing. *Nature Biotechnology*, *26*, 1135–1145.
34. Smyth, G. K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, *3*, Article 3.
35. Song, J. S., et al. (2007). Microarray blob-defect removal improves array analysis. *Bioinformatics*, *23*, 966–971.
36. Vokes, S. A., et al. (2007). Genomic characterization of Gli-activator targets in sonic hedgehog-mediated neural patterning. *Development*, *134*, 1977–1989.
37. Vokes, S. A., Ji, H., Wong, W. H., & McMahon, A. P. (2008). A genome-scale analysis of the cis-regulatory circuitry underlying sonic hedgehog mediated patterning of the mammalian limb. *Genes & Development*, *22*, 2651–2663.
38. Wu, Z., Irizarry, R. A., Gentleman, R., et al. (2004). A model based background adjustement for oligonucleotide expression arrays. *Journal of the American Statistical Association*, *99*, 909–917.
39. Zheng, M., Barrera, L. O., Ren, B., Wu, & Y. N. (2007). ChIP-chip: Data, model, and analysis. *Biometrics*,*63*, 787–796.

Chapter 13
eQTL Mapping for Functional Classes of *Saccharomyces cerevisiae* Genes with Multivariate Sparse Partial Least Squares Regression

Dongjun Chung and Sündüz Keleş

Abstract The availability of high-throughput genotyping technologies and microarray assays has enabled investigation of genetic variations that influence levels of gene expression. Expression Quantitative Trait Loci (eQTL) mapping methods have been successfully used to identify the genetic basis of gene expression which in turn led to identification of candidate genes and construction of regulatory networks. One challenging statistical aspect of eQTL mapping is the existence of thousands of traits. We have recently proposed a multivariate sparse partial least squares framework for mapping multiple quantitative traits and identifying genetic variations that affect the expression of a group of genes. In this book chapter, we provide a comprehensive illustration of this methodology with a *Saccharomyces cerevisiae* linkage study. Data from this study involves segregants from a cross between two *Saccharomyces cerevisiae* strains. Our application focuses on elucidating genomic markers that affect expression of functional yeast gene classes. We illustrate identification of eQTL regions affecting whole functional classes of genes as well as eQTL regions influencing individual genes.

13.1 Introduction

Expression quantitative trait loci (eQTL) mapping aims to identify which markers are linked to transcripts with detectable effects and to estimate the magnitudes of these effects. A number of recent studies demonstrated utility of eQTL mapping in a broad range of biological applications [7,11,31,32,34,38,40,46,49] and the emerging field of "Genetical Genomics" has created much excitement and enthusiasm

S. Keleş (✉)
Department of Statistics and Department of Biostatistics and Medical Informatics,
University of Wisconsin, 53706, Madison, WI, USA
e-mail: keles@stat.wisc.edu

D. Chung
Department of Statistics, University of Wisconsin, 53706, Madison, WI, USA
e-mail: chungdon@stat.wisc.edu

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_13,
© Springer-Verlag Berlin Heidelberg 2011

[33]. Statistical challenges of eQTL mapping have been immediately realized [26]. Initial methods for eQTL mapping were usually either transcript-specific or marker-specific analyses [27] and lacked proper control of Type-I error rate. Recent efforts for eQTL mapping focused on combined analysis of all the transcript and marker data by collapsing these two approaches. In particular, most recent advancements in statistical methods for eQTL mapping aim to properly account for multiplicities across the genomic markers and transcripts and correlations among transcripts. Some of these approaches are by [9, 18, 21, 27]. A distinct feature of eQTL analysis is the existence of multi-traits. It is well known from the traditional QTL literature that repeated application of single trait analysis is not optimal since information in the correlation structure among the traits is not utilized [2, 22]. To address the issue of multi-traits in eQTL mapping, we formulated the eQTL mapping problem as a variable selection problem in a multivariate response regression in a recent work [12]. The multivariate response setting is facilitated by clustering genes across the segregating population of individuals prior to eQTL mapping. We then utilized sparse partial least squares [13] as a simultaneous variable selection and dimension reduction approach to identify linkages for each cluster. This framework, named as M-SPLS eQTL, offers a computationally fast alternative for analyzing multiple transcript and marker data simultaneously for gaining power and avoiding multiplicities for good error control. BAYES approach of [21] utilizes all the transcripts simultaneously as multivariate traits for eQTL mapping. In contrast, M-SPLS eQTL mapping provides a compromise between individual transcript-based analysis and analysis by using all the traits simultaneously.

Current state of the art for interpreting identified eQTL and further hypothesis generation often relies on a version of gene set enrichment analysis whereby genes those map to same locus are tested for enrichment in functional gene sets [48]. Since functional gene sets readily partition the set of transcripts into functionally meaningful groups, they define a clustering of the genes. Therefore, utilizing genes within functional groups in a multi-trait analysis might provide an alternative to clustering genes. In this chapter, we focus on utilizing functional gene classes in M-SPLS eQTL analysis. We start our exposition by providing a brief overview of the eQTL mapping with multivariate sparse partial least squares regression [12]. Then, we demonstrate step-by-step how M-SPLS eQTL identifies many biologically relevant eQTL when we utilize functional gene classes in a yeast eQTL study by [5].

13.2 eQTL Mapping with M-SPLS

In ill-conditioned linear regression problems, partial least squares (PLS) regression is often utilized as an alternative to ordinary least squares regression. Although PLS had been traditionally promoted for regression problems with a large number of variables, [13] showed that PLS suffers from the curse of dimensionality in the contemporary large p, small n setting. [13] developed a sparse version of PLS (SPLS) to achieve accurate prediction and variable selection simultaneously by producing sparse linear combinations of the original predictors. SPLS regression can select a higher number of relevant variables than the available sample size and it can handle multivariate response without additional computational complexity. SPLS has two tuning parameters, K and η, which are the number of latent components and the sparsity parameter, respectively. K is constrained to be smaller than the rank of the predictor matrix and η has a value between 0 and 1. As K gets smaller and η approaches 1, SPLS results in sparse solutions. Further details about SPLS can be found at [13].

Chun and Keleş demonstrated that SPLS has several attractive properties that motivate its use for eQTL mapping [12, 13]. Many recent eQTL mapping studies suggest that most eQTL have weak effects and many transcripts require multiple loci (markers) under additive models [5]. Utilizing expression of a group of genes as multivariate response, SPLS facilitates joint analysis of multiple transcripts and markers to boost weak linkage signals. High correlations among markers in close proximity can be taken into account due to the ability of SPLS to handle correlated covariates. M-SPLS eQTL consists of the following three steps.

1. *Forming multi-traits.* This first step forms "biologically interesting" clusters of gene expression which are then viewed as multi-traits. Clustering genes based on their expression profiles among experimental units to identify correlated group of genes is one way of forming such biologically interesting groups. There is a rich literature on clustering of gene expression data [23] and the choice of clustering method in a given experiment relies on multiple factors concerning the design of the experiment. As discussed in the introduction, utilizing pre-defined functional classes of genes is an alternative way of forming biologically meaningful groups of genes.
2. *Marker selection with M-SPLS regression.* The second step considers the expression values within a cluster of genes as a multivariate response and forms a cluster-level multivariate response regression. For each cluster k, we define a G_k-dimensional response vector $Y_{i.}^{(k)}$ to denote the expressions of all the G_k genes, measured on the i-th subject and $X_{i.}$ to denote the marker genotype vector for the same i-th subject. We then consider a cluster-specific marker model

$$Y_{i.}^{(k)} = X_{i.}B^{(k)} + E_{i.},$$

   where $E_{i.}$ denotes the random error matrix and $B^{(k)}$ is a $p \times G_k$ matrix representing the contribution of each marker $m \in \{1, \dots, p\}$ to the expression variation of each transcript $g \in \{1, \dots, G_k\}$ of cluster k. Such a linear model is fitted for every cluster using the SPLS regression. This step identifies markers affecting all or a subgroup of genes within the cluster.
3. *Transcript selection with bootstrap confidence intervals.* Since SPLS estimates effects of the selected markers on all of the genes in the cluster simultaneously, it can result in some false linkages. We construct bootstrap confidence intervals for each marker (selected in step 2 above) and transcript combination to screen out linkages of transcripts with small effects. The final summary of linkages

contains marker/transcript combinations for which the confidence intervals do not include zero.

`R` package `spls` provides an implementation of SPLS regression and it can be downloaded from the Comprehensive R Archive Network (CRAN) at http://cran.r-project.org/web/packages/spls. This package provides functions for tuning and fitting of SPLS and functions specific for eQTL mapping. In what follows, `spls` version 2.1-0 was used.

13.3 Data Description and Preprocessing

We provide an illustration of the M-SPLS eQTL using the eQTL data of 112 yeast segregants generated from two *Saccharomyces cerevisiae* parent strains, BY4716 and RM11-1a (BY and RM for short, respectively) [5]. BY is a standard laboratory strain and RM is a wild isolate from a California vineyard. Expression levels of 6,229 genes and genotypes of 2,956 SNPs were measured in each of the segregants. The genotype profile of each marker is binary, indicating the parental strain origin of the allele. Ambiguous origin was set as missing value.

On average, 0.97% proportion of expression values and 1.89% proportion of genotype values were missing for each gene and marker, respectively. There were no missing values for 4,474 genes and 787 markers. We imputed the missing expression values using a $k = 15$ nearest neighbor method [44] as in [6]. Missing genotype data were imputed using the hidden Markov model approach implemented in function `fill.geno` of R package `R/qtl` [8]. We excluded genes with more than 10% missing segregant expression and excluded segregants with more than 200 missing markers. Finally, we combined nearby SNPs with the exact same genotype across all the segregants. This resulted in 1,028 markers and 6,089 genes for 105 segregants.

In the first step of M-SPLS eQTL, we utilized functional classes of yeast genes developed using *Saccharomyces* Genome Database (SGD) [14] as clusters of interest. These functional classes were used before, for example in [16]. We focused our attention on 44 classes with <100 members. The list of these 44 classes is provided in Table 13.1.

13.4 Results

13.4.1 eQTL Mapping of the 'Mating' Class

We first focus on the Mating functional class and identify markers that contribute to variability of transcript levels among the yeast segregants for 25 genes of this class. These genes are: AGA1, AGA2, STE6, STE14, MFA2, MFA1, STE23, STE3, MF(ALPHA)2, MF(ALPHA)1, SAG1, BAR1, STE2, AFR1, FIG1, FIG2, SST2, MOT2, MOT3, PEA2, OPY1, FUS3, SSF1, FIG4, and MID2. In the following `R` script, the $105 \times 1,028$ matrix `x` contains genotype data and the 105×25 matrix

Table 13.1 Summary of eQTL mapping for 44 functional gene classes. Denominator in each entry of the table denotes the number of selected markers for each functional class

Class (No. of genes)	No. of selected markers	*cis*-acting markers	Markers close to genes	Markers within genes	Markers with one linkage
Arginine (15)	1	0 / 1	1 / 1	1 / 1	0 / 1
Asparagine (7)	1	1 / 1	1 / 1	1 / 1	0 / 1
Autophagy (9)	2	0 / 2	2 / 2	2 / 2	0 / 2
Biotin (4)	1	0 / 1	1 / 1	1 / 1	0 / 1
DrugResistance (12)	109	1 / 109	109 / 109	102 / 109	41 / 109
Endocytosis (19)	5	0 / 5	5 / 5	5 / 5	0 / 5
FattyAcid (35)	21	3 / 21	21 / 21	20 / 21	0 / 21
Flavin (5)	7	0 / 7	7 / 7	7 / 7	0 / 7
Flocculation (5)	5	1 / 5	5 / 5	5 / 5	4 / 5
Galactose (6)	124	2 / 124	124 / 124	105 / 124	67 / 124
GeneralTFs (33)	72	2 / 72	72 / 72	58 / 72	22 / 72
Glutamate (5)	7	0 / 7	7 / 7	7 / 7	0 / 7
Glutathione (7)	11	0 / 11	11 / 11	10 / 11	3 / 11
Glycerol (7)	103	0 / 103	103 / 103	92 / 103	17 / 103
Glycogen (8)	195	2 / 195	194 / 195	164 / 195	18 / 195
Heme (8)	8	3 / 8	8 / 8	8 / 8	1 / 8
Histidine (8)	80	6 / 80	80 / 80	72 / 80	2 / 80
InvasiveGrowth (10)	96	2 / 96	96 / 96	89 / 96	49 / 96
Isoleucine (7)	13	2 / 13	13 / 13	11 / 13	0 / 13
Leucine (5)	2	1 / 2	2 / 2	2 / 2	0 / 2
Lysine (9)	6	3 / 6	6 / 6	5 / 6	0 / 6
Maltose (4)	89	3 / 89	89 / 89	82 / 89	0 / 89
Mating (25)	2	0 / 2	2 / 2	1 / 2	0 / 2
Mitochondria (45)	10	3 / 10	10 / 10	8 / 10	0 / 10
MitochondrialRPs (54)	20	1 / 20	20 / 20	17 / 20	0 / 20
NuclearProteinTargetting (50)	220	12 / 220	220 / 220	187 / 220	44 / 220
PentosePhosphate (9)	7	0 / 7	7 / 7	7 / 7	0 / 7
Peroxisome (20)	5	3 / 5	5 / 5	5 / 5	0 / 5
Phospholipid (35)	26	2 / 26	26 / 26	23 / 26	0 / 26
Polyamine (5)	4	0 / 4	4 / 4	3 / 4	0 / 4
Proline (4)	11	0 / 11	11 / 11	9 / 11	8 / 11
Proteasome (30)	57	3 / 57	57 / 57	53 / 57	0 / 57
Purine (23)	43	1 / 43	43 / 43	39 / 43	6 / 43
Pyrimidine (23)	19	1 / 19	19 / 19	18 / 19	1 / 19
RNAPol (55)	75	4 / 75	75 / 75	64 / 75	0 / 75
rRNAprocessing (36)	3	0 / 3	3 / 3	3 / 3	0 / 3
sphingolipids (12)	31	6 / 31	31 / 31	28 / 31	15 / 31
Sterols (27)	13	1 / 13	13 / 13	11 / 13	0 / 13
TFs (18)	5	1 / 5	5 / 5	5 / 5	0 / 5
Thiamine (11)	197	0 / 197	197 / 197	170 / 197	74 / 197
tRNAprocessing (32)	172	5 / 172	172 / 172	150 / 172	18 / 172
tRNAsynth (30)	3	0 / 3	3 / 3	2 / 3	0 / 3
Tryptophan (12)	15	0 / 15	15 / 15	14 / 15	0 / 15
Vacuole (66)	26	4 / 26	26 / 26	25 / 26	0 / 26

`y.sel` is the matrix of expression values of 25 genes in the Mating class, across the yeast segregants. The function `cv.spls` searches optimal tuning parameters, K and eta, that minimize cross-validated mean squared prediction error within the range that user specifies. eta should have a value between 0 and 1. K is integer valued and can range between 1 and $\min\{p, (v-1)n/v\}$, where p is the number of predictors, n is the sample size, and v is the number of cross-validation folds. For example, if ten-fold cross-validation is used (default), K will be smaller than $min\{p, 0.9n\}$. Here, we search K between 1 and 10 and eta between 0.01 and 0.99.

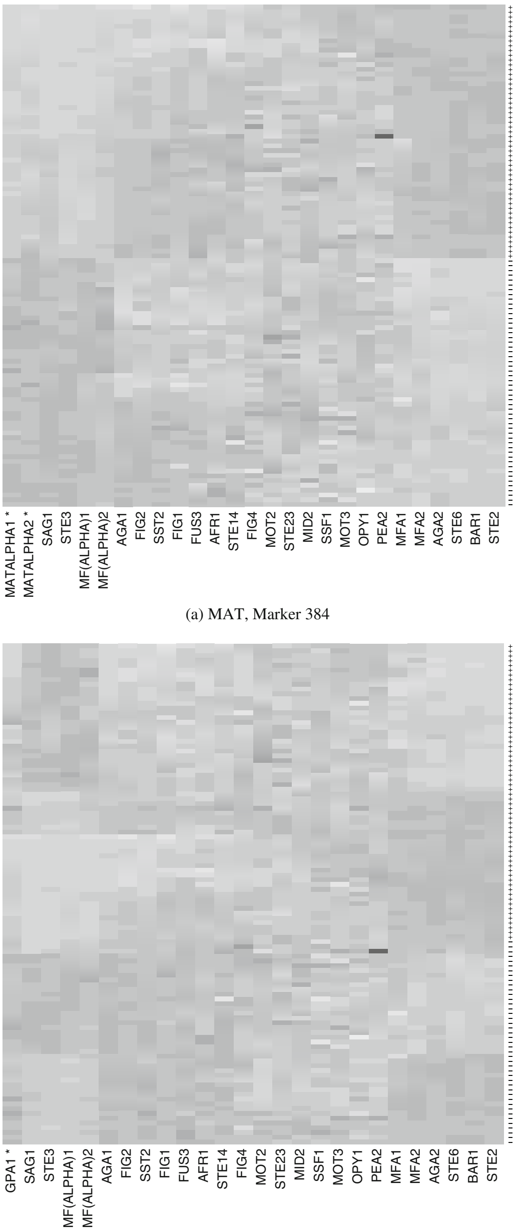
```
> cvs <- cv.spls( x, y.sel, K = c(1:10),
        eta = seq(0.01, 0.99, 0.01) )
```

For the Mating class, the optimal parameters are determined as $K = 2$ and $eta = 0.99$. The final SPLS fit can be obtained as follows.

```
> fits <- spls( x, y.sel, K=cvs$K.opt, eta=cvs$eta.opt )
> fits

Sparse Partial Least Squares for multivariate responses
----
Parameters: eta = 0.99, K = 2, kappa = 0.5
PLS algorithm:
pls2 for variable selection, simpls for model fitting

SPLS chose 2 variables among 1028 variables

Selected variables:
384     1248
```

M-SPLS regression on the Mating functional class selects two (combined) markers (384 and 1,248) with *trans*-effects. Combined marker 384 consists of 2 markers with the identical genotype, located at positions 201, 166 and 201, 167 bp in chromosome 3. Combined marker 1,248 consists of 7 markers with the same genotype and they are located at positions 111, 679–111, 690 bp in chromosome 8. Marker 384 resides within the MAT locus and it is located 992 bp upstream of MATALPHA2 gene and 201 bp downstream of MATALPHA1 gene. Marker 1,248 resides 1,804 bp downstream of GPA1 gene. These loci were identified to influence the Mating class genes in a previous study [6]. In particular, it is known that the MAT locus determines mating type of a haploid yeast cell by the genetic composition of the locus and the polymorphism in GPA1 affects expression of pheromone response genes [6].

Transcription factor (TF) MCM1 can bind to both the α- and a-specific genes and products of MAT determine whether transcriptional activation takes place [20]. Panel (a) of Fig. 13.1 shows the heatmap of the gene expression of the MAT genes (MATALPHA1 and MATALPHA2) and the genes in the Mating class. MAT genes are differentionally expressed for two different genotypes of Marker 384 and both MATALPHA1 and MATALPHA2 are overexpressed in BY strains and



(a) MAT, Marker 384

(b) GPA1, Marker 1248

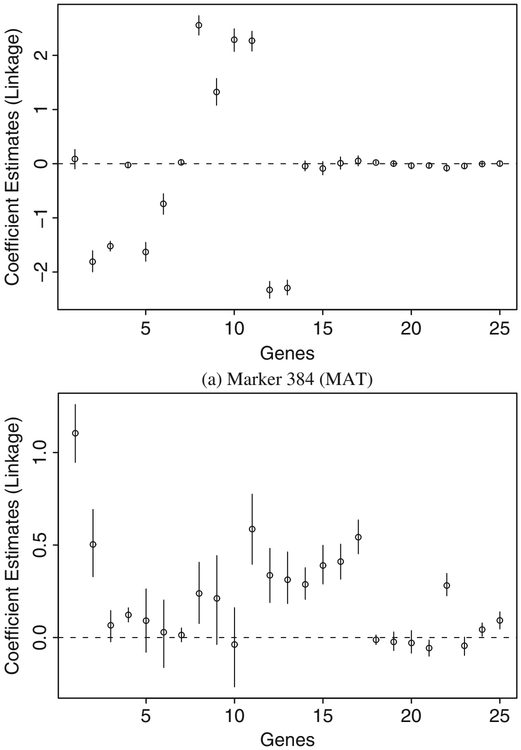
Fig. 13.1 (**a**) Heatmap of the gene expression of MAT genes (*first two columns*; marked with '*') and the genes in the Mating class. Rows are grouped by genotype of Marker 384, which resides upstream of MATALPHA2 gene and downstream of MATALPHA1 gene. (**b**) Heatmap of the gene expression of GPA1 gene (*first column*; marked with '*') and the genes in the Mating class. Rows are grouped by the genotype of Marker 1,248, which resides downstream of GPA1 gene. Bright and dark color scheme indicate over- and under-expression, respectively. '+' and '−' represent genotypes 'BY' and 'RM', respectively underexpressed in RM strains. The overexpression of MATALPHA1 ‘turns on’ the α-specific genes such as MF(ALPHA)1, MF(ALPHA)2, STE3, and SAG1 while the overexpression of MATALPHA2 ‘turns off’ the a-specific genes such as MFA1, MFA2, STE2, STE6, BAR1, and AGA2 [20].

Panel (b) of Fig. 13.1 displays the heatmap of the gene expression of GPA1 gene and the genes in the Mating class. Different sets of genes in the class are differentially expressed for two different genotypes of Marker 1,248, e.g., FIG1, FIG2, AGA1, SST2, FUS3, and AFR1. [49] suggested that the polymorphism of GPA1 may affect its binding to STE2 and STE3 and this again changes the expression of genes in the Mating class. However, [41] discusses that STE2 and STE3 are not co-expressed with GPA1 and they are more affected by MAT than GPA1. [41] suggested that the gene expression of STE12 is linked to GPA1 locus by considering the biological knowledge that signals initiated from GPA1, STE2, and STE3 propagate through the MAPK signaling cascade that reach STE12 [47]. The genes differentially expressed for two different genotypes of Marker 1,248 are target genes of the transcription factor STE12 according to the transcription factor binding data of [19] and this may suggest that such differential expression is related to the activity of TF STE12.

Next, we further investigate the linkages of these two markers with the genes in the class. `ci.spls` function constructs bootstrap confidence intervals for each marker/transcript combination (default is 95% confidence intervals). `correct.spls` provides refined linkages by setting the marker/transcript combinations to zero if their corresponding confidence intervals include zero. `fits$A` contains the index of selected markers and the final summary of linkages can be obtain by the command `cf[ fits$A, ]`.

```
> ci.f <- ci.spls( fits )
> cf <- correct.spls( ci.f )
> cf[ fits$A, ]
         AGA1       AGA2      STE6     STE14      MFA2
384  0.000000 -1.8098963 -1.522525 0.0000000 -1.629656
1248 1.104209  0.5030974  0.000000 0.1220773  0.000000
           MFA1 STE23      STE3 MF(ALPHA)2 MF(ALPHA)1
384  -0.7407644     0 2.5571961      1.324   2.288917
1248  0.0000000     0 0.2383098      0.000   0.000000
          SAG1       BAR1       STE2      AFR1      FIG1
384  2.2701921 -2.3299790 -2.2942640 0.0000000 0.0000000
1248 0.5859102  0.3367588  0.3124113 0.2873071 0.3895847
          FIG2      SST2 MOT2 MOT3 PEA2
384  0.0000000 0.0000000    0    0    0
1248 0.4105893 0.5426099    0    0    0
            OPY1        FUS3 SSF1       FIG4       MID2
384   0.00000000 -0.08039047    0 0.00000000 0.00000000
1248 -0.05678439  0.28145494    0 0.04289189 0.09243976
```

Construction of confidence intervals reveals that genes MFA1, MFA2, MF (ALPHA)1, MF(ALPHA)2, and STE6 have linkages only with Marker 384, whereas, genes FIG1, FIG2, FIG4, AGA1, STE14, AFR1, SST2, OPY1, and MID2 have linkages only with Marker 1,248. In contrast, AGA2, SAG1, BAR1, STE2, STE3, and FUS3 have linkages with both of markers 384 and 1,248. Linkages with MAT are stronger in AGA2, SAG1, BAR1, STE2, and STE3, whereas linkage with GPA1 is stronger in FUS3. STE23, MOT2, MOT3, PEA2, and SSF1 do not have linkages with either of the markers. Figure 13.2 displays the 95% confidence intervals of coefficient estimates for markers 384 and 1,248, respectively, and provides visual summary of the linkages.



(a) Marker 384 (MAT)

(b) Marker 1248 (GPA1)

Fig. 13.2 Ninety-five percent confidence intervals of coefficient estimates for markers 384 and 1,248. Marker 384 resides upstream of MATALPHA2 gene and downstream of MATALPHA1 gene. Marker 1,248 resides downstream of GPA1 gene. Circles and solid vertical lines indicate point estimates and their corresponding confidence intervals, respectively. Numbers 1–25 along the x-axis correspond to the genes in the Mating class, AGA1, AGA2, STE6, STE14, MFA2, MFA1, STE23, STE3, MF(ALPHA)2, MF(ALPHA)1, SAG1, BAR1, STE2, AFR1, FIG1, FIG2, SST2, MOT2, MOT3, PEA2, OPY1, FUS3, SSF1, FIG4, and MID2

13.4.2 Summary of the eQTL Analysis for all the Functional Classes

In this section, we summarize results from the analysis of all the 44 functional classes by the M-SPLS eQTL. As in the previous section, we searched K between 1 and 10 and eta between 0.01 and 0.99. Table 13.1 displays a summary of the results. For 13 of the functional classes, more than 50 markers were selected. There are two factors contributing to such high number of selected markers. Functional classes with large number of genes and small average within-cluster correlation tend to have a large number of selected markers (Panels (a) and (b) of Fig. 13.3). This is expected since if the transcript levels of genes within the functional category are not correlated, it is reasonable to argue that different sets of markers are contributing to expression variation in individual genes. Therefore, as the number of uncorrelated genes within the cluster increases, the number of selected markers will tend to increase. We also observe some functional groups with a small number of genes and large number of selected markers, for example, Galactose, Glycerol, Thiamine, and DrugResistance, and within-cluster correlation for these groups are quite small ($<$0.2).

For each functional class, we determined whether the selected markers are *cis*- or *trans*-acting (column 2 in Table 13.1). Following [7] and [49], a marker is labeled as *cis*-acting if any of the functional class genes that it has linkage with is within 10 kb of the marker. Consistent with the results of previous eQTL studies [49], our results indicate that most linkages are in *trans*. However, 64% of the functional classes (28 out of 44) have at least one *cis*-acting marker. We further checked whether each selected marker is within 2 kb of any known yeast genes (column 3) and whether the marker is within the coding region of any known yeast genes (column 4). In order to assess whether markers have broad effects on many genes in the class or specific effects on a small subset of the genes, we checked the number of markers that have linkage with only one gene in the class (column 5). In 61% of the functional classes (27 out of 44), all markers have linkages with more than one gene in the class.

Further analysis of each gene category revealed that many gene clusters are linked to markers that reside within or in close proximity of known yeast genes (Columns 3 and 4 of Table 13.1). Arginine functional category genes that are responsible for Arginine biosynthesis are linked to a marker that resides at position $76,127$ bp on chromosome 3. This marker is within the coding region of the AGP1 gene which is an arginine/glutamate permease gene [39]. Asparagine functional class has an eQTL region that spans positions $468,981$ bp to $489,760$ bp on chromosome 12. This region consists of 30 markers with the same genotype and some of these markers exhibit *cis*-acting property. A subset of the markers reside exactly on the coding region of the ASP3-1 gene and some are within 2 Kb of ASP3-2 gene. Both of these genes are members of the Asparagine functional group and are known to be involved in asparagine catabolism [29]. Another example of a *cis*-acting marker is at position $460,945$ bp of chromosome 7. This marker is 726 bp upstream of the KAP122 which is a member of the DrugResistance functional class.

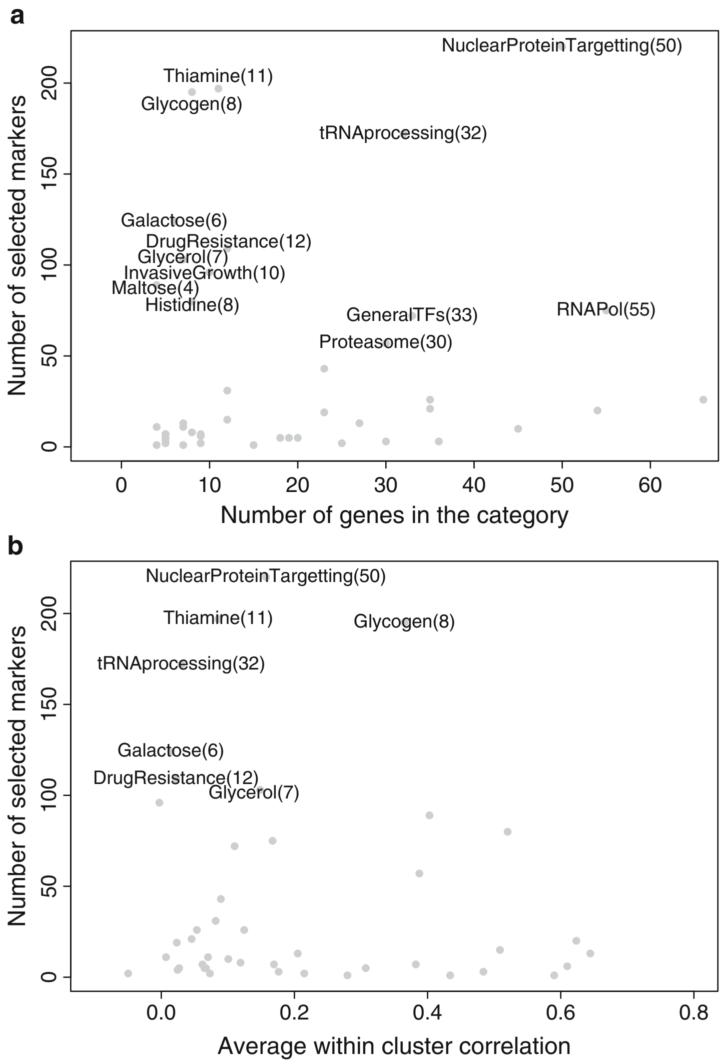


Fig. 13.3 (**a**) Numbers of selected markers vs. number of genes in the category. Class names are provided for classes with more than 50 selected markers. (**b**) Numbers of selected markers vs. average within-cluster correlation. Class names are provided for classes with more than 100 selected markers. Number within parenthesis is number of genes in each functional class

This gene might play a role in regulation of pleiotropic drug resistance [10]. Two other markers linking to this category are in the vicinity of other drug resistance related genes AQR1 [43, 45] and DTR1 [15, 17].

For the FattyAcids category, a marker at position 51, 324 bp of chromosome 1 has linkage with 12 of the 35 genes within this category and resides within the coding region of the OAF1 gene. OAF1 is a Oleate-activated transcription factor and it activates genes involved in beta-oxidation of fatty acids [24, 36]. Markers at positions 584, 351 and 584, 357 bp in chromosome 2 with the identical genotype have linkage with 6 of the 35 genes in FattyAcids category. These markers are located at 445 bp downstream of the EHT1 gene. EHT1 is ethanol O-acyltransferase that plays a minor role in medium-chain fatty acid ethyl ester biosynthesis [37].

Galactose functional category has two *cis*-acting markers that reside upstream of the GAL3 gene involved in the activation of the GAL genes in response to galactose [4, 35]. These markers are located at positions 463, 264 and 463, 267 bp on chromosome 4. A *trans*-acting marker at position 76, 127 bp of chromosome 3 resides within the AGP1 gene which is involved in uptake of asparagine, glutamine, and other amino acids [39]. This marker has linkages with 4 of the 5 genes in the functional class Glutamate.

Heme class has a *cis*-acting marker that resides upstream of the HEM3 gene involved in the third step in heme biosynthesis [28]. This marker is located at position 95, 437 bp in chromosome 4. Genes in histidine category have linkages with two *cis*-acting markers, located at position 64, 311 bp in chromosome 3 and position 141, 014 bp in chromosome 9. These two markers are placed 1,623 bp downstream of HIS4 and 1,911 bp upstream of HIS5 genes, respectively. HIS4 is involved in the second, third, ninth and tenth steps in histidine biosynthesis [25] while HIS5 is involved in the seventh step [1].

Three among five genes in the leucine category (LEU1, LEU2, and LEU4) have linkages with Marker 356 consisting of 21 markers with the identical genotype, located at positions 90, 412–92, 391 bp in chromosome 3. 13 of these markers reside within the coding region of LEU2 and the other 8 markers reside upstream of LEU2. LEU2 is involved in the third step in the leucine biosynthesis pathway [3]. Such linkages were also found in previous studies and LEU2 was considered as the putative regulator of the leucine category [7, 42]. Figure 13.4 shows the heatmap of the gene expression of the LEU2 gene and the genes in the leucine category. LEU2 is not expressed in RM strain because RM strain is LEU2-deleted [41, 49]. LEU1 and LEU4 are overexpressed in RM strain and this suggests a potential compensation effect due to the loss of LEU2 in RM strain [41, 42]. The transcription factor (TF) LEU3 binds to LEU2 [19, 30] and [41] argued that the genetic variation in LEU2 perturbs the TF activity of LEU3 and the perturbed TF activity affects the expression of genes in the leucine category.

13.5 Conclusion

We provided an application of M-SPLS eQTL to analyze eQTL data for 44 yeast functional gene classes. In particular we focused on the `R` package `spls` that facilitates the application of sparse partial least squares regression for eQTL mapping. We have illustrated that this approach provides a principled way of browsing through linkages generated by eQTL mapping and the focus on functional gene categories helps to generate hypotheses for further investigation.

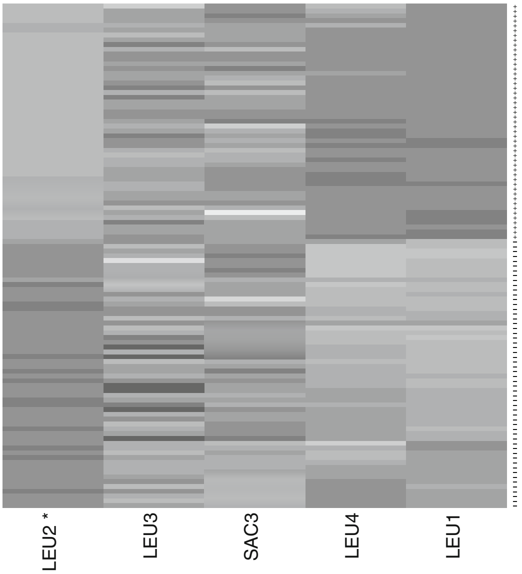


Fig. 13.4 Heatmap of the gene expression of LEU2 (*first column*; marked with '*') and the genes in the leucine category. Rows are grouped by the genotype of Marker 356, which resides within LEU2. Bright and dark color scheme indicate over- and under-expression, respectively. '+' and '−' represent genotypes 'BY' and 'RM', respectively

Acknowledgements This research was supported by NSF grant DMS 0804597 to SK.

References

1. Alifano, P., Fani, R., Liò, P., Lazcano, A., Bazzicalupo, M., Carlomagno, M., & Bruni, C. (1996). Histidine biosynthetic pathway and genes: Structure, regulation, and evolution. *Microbiological Reviews*, *60*, 44–69.
2. Allison, D. B., Thiel, B., Jean, P. S., Elston, R. C., Infante, M. C., & Schork, N. J. (1998). Multiple phenotype modeling in gene-mapping studies of quantitative traits: Power advantages. *American Journal of Human Genetics*, *63*, 1190–1201.
3. Andreadis, A., Hsu, Y., Hermodson, M., Kohlhaw, G., & Schimmel, P. (1984). Yeast LEU2. Repression of mRNA levels by leucine and primary structure of the gene product. *The Journal of Biological Chemistry*, *259*, 8059–8062.
4. Bhat, P., & Murthy, T. (2001). Transcriptional control of the GAL/MEL regulon of yeast *Saccharomyces cerevisiae*: Mechanism of galactose-mediated signal transduction. *Molecular Microbiology*, *40*, 1059–1066.
5. Brem, R., & Kruglyak, L. (2005). The landscape of genetic complexity across 5,700 gene expression traits in yeast. *Proceedings of the National Academy of Sciences of the United States of America*, *102*, 1572–1577.
6. Brem, R., Storey, J., Whittle, J., & Kruglyak, L. (2005). Genetic interactions between polymorphisms that affect gene expression in yeast. *Nature*, *436*, 701–703.
7. Brem, R., Yvert, G., Clinton, R., & Kryglyak, L. (2002). Genetic dissection of transcriptional regulation in budding yeast. *Science*, *296*, 752–755.
8. Broman, K. W., Wu, H., Sen, S., & Churchill, G. A. (2003). R/qtl: QTL mapping in experimental crosses. *Bioinformatics*, *19*, 889–890.
9. Chen, M., & Kendziorski, C. (2007). A statistical framework for expression quantitative trait loci (eQTL) mapping. *Genetics*, *177*, 761–771.

10. Chen, W., Balzi, E., Capieaux, E., Choder, M., & Goffeau, A. (1991). The DNA sequencing of the 17 kb HindIII fragment spanning the LEU1 and ATE1 loci on chromosome VII from *Saccharomyces cerevisiae* reveals the PDR6 gene, a new member of the genetic network controlling pleiotropic drug resistance. *Yeast*, *7*, 287–299.
11. Chesler, E. J., Lu, L., Shou, S., Qu, Y., Gu, J., Wang, J., Hsu, H. C., Mountz, J. D., Baldwin, N. E., Langston, M. A., Threadgill, D. W., Manly, K. F., & Williams, R. W. (2005). Complex trait analysis of gene expression uncovers polygenic and pleiotropic networks that modulate nervous system function. *Nature Genetics*, *37*(4), 233–242.
12. Chun, H., & Keleş, S. (2009). Expression quantitative trait loci mapping with multivariate sparse partial least squares regression. *Genetics*, *182*, 79–90.
13. Chun, H., & Keleş, S. (2010). Sparse partial least squares for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B*, *72*, 3–25.
14. Dwight, S. S., Balakrishnan, R., Christie, K. R., Costanzo, M. C., Dolinski, K., Engel, S. R., Feierbach, B., Fisk, D. G., Hirschman, J., Hong, E. L., Issel-Tarver, L., Nash, R. S., Sethuraman, A., Starr, B., Theesfeld, C. L., Andrada, R., Binkley, G., Dong, Q., Lane, C., Schroeder, M., Weng, S., Botstein, D., & Cherry, J. M. (2004). *Saccharomyces* genome database: Underlying principles and organisation. *Briefings in Bioinformatics*, *5*(1), 922. URLhttp://dx.doi.org/10.1093/bib/5.1.9
15. Felder, T., Bogengruber, E., Tenreiro, S., Ellinger, A., Sá-Correia, I., & Briza, P. (2002). Dtrlp, a multidrug resistance transporter of the major facilitator superfamily, plays an essential role in spore wall maturation in *Saccharomyces cerevisiae*. *Eukaryotic cell*, *1*, 799–810.
16. Gasch, A., & Eisen, M. (2002). Exploring the conditional coregulation of yeast gene expression through fuzzy k-means clustering. *Genome Biology*, *3*, research0059.1”0059.22.
17. Gbelska, Y., Krijger, J., & Breunig, K. (2006). Evolution of gene families: The multidrug resistance transporter genes in five related yeast species. *FEMS Yeast Research*, *6*, 345–355.
18. Gelfond, J. A. L., Ibrahim, J. G., & Zou, F. (2007). Proximity model for expression quantitative trait loci (eQTL) detection. *Biometrics*, *63*(4), 1108–1116.
19. Harbison, C., Gordon, D., Lee, T., Rinaldi, N., Macisaac, K., Danford, T., Hannett, N., Tagne, J. B., Reynolds, D., Yoo, J., Jennings, E., Zeitlinger, J., Pokholok, D., Kellis, M., Rolfe, P., Takusagawa, K., Lander, E., Gifford, D., Fraenkel, E., & Young, R. (2004). Transcriptional regulatory code of a eukaryotic genome. *Nature*, *431*, 99–104.
20. Herskowitz, I. (1989). A regulatory hierarchy for cell specialization in yeast. *Nature*, *342*, 749–757.
21. Jia, Z., & Xu, S. (2007). Mapping quantitative trait loci for expression abundance. *Genetics*, *176*, 611–623.
22. Jiang, C., & Zeng, Z. (1995). Multiple trait analysis of genetic mapping for quantitative trait loci. *Genetics*, *140*, 1111–1127.
23. Jiang, D., Tang, C., & Zhang, A. (2004). Cluster analysis for gene expression data: A survey. *IEEE Transactions on Knowledge and Data Engineering*, *16*(11), 1370–1386.
24. Karpichev, I. V., & Small, G. M. (1998). Global regulatory functions of Oaf1p and Pip2p (Oaf2p), transcription factors that regulate genes encoding peroxisomal proteins in *Saccharomyces cerevisiae*. *Molecular Cell Biology*, *18*, 6560–6570.
25. Keesey, J. J., Bigelis, R., & Fink, G. (1979). The product of the his4 gene cluster in *Saccharomyces cerevisiae*. a trifunctional polypeptide. *The Journal of Biological Chemistry*, *254*, 7427–7433.
26. Kendziorski, C., & Wang, P. (2006). A review of statistical methods for expression quantitative trait loci mapping. *Mammalian Genome*, *17*(6), 509–517.
27. Kendziorski, C. M., Chen, M., Yuan, M., Lan, H., & Attie, A. D. (2006). Statistical methods for expression quantitative trait loci (eQTL) mapping. *Biometrics*, *62*, 19–27.
28. Keng, T., Richard, C., & Larocque, R. (1992). Structure and regulation of yeast HEM3, the gene for porphobilinogen deaminase. *Molecular and General Genetics*, *234*, 233–243.
29. Kim, K. W., Kamerud, J. Q., Livingston, D. M., & Roon, R. (1988). Asparaginase II of *Saccharomyces cerevisiae*. Characterization of the ASP3 gene. *Journal of Biological Chemistry*, *263*, 11948–11953.

30. Kohlhaw, G. (2003). Leucine biosynthesis in fungi: Entering metabolism through the back door. *Microbiology and Molecular Biology Reviews*, *67*, 1–15.
31. Lan, H., Chen, M., Flowers, J. B., Yandell, B. S., Stapleton, D. S., Mata, C. M., n Keen Mui, E. T., Flowers, M. T., Schueler, K. L., Manly, K. F., Williams, R. W., Kendziorski, C., & Attie, A. D. (2006). Combined expression trait correlations and expression quantitative trait locus mapping. *PLoS Genetics*, *2*, e6.
32. Lan, H., Stoehr, J. P., Nadler, S. T., Schueler, K. L., Yandell, B. S., & Attie, A. D. (2003). Dimension reduction for mapping mRNA abundance as quantitative traits. *Genetics*, *164*, 1607–1614.
33. Li, J., & Burmeister, M. (2005). Human molecular genetics review issue 2. *BMC Genomics*, *14*(2), R163–R169.
34. Morley, M., Molony, C. M., Weber, T. M., Devlin J. L. snd Ewens, K. G., Spielman, R. S., & Cheung, V. G. (2004). Genetic analysis of genome-wide variation in human gene expression. *Nature*, *430*(7001), 743–747.
35. Platt, A., & Reece, R. (1998). The yeast galactose genetic switch is mediated by the formation of a Gal4p-Gal80p-Gal3p complex. *The EMBO Journal*, *17*, 4086–4091.
36. Rottensteiner, H., Kal, A. J., Hamilton, B., Ruis, H., & Tabak, H. F. (1997). A heterodimer of the Zn2Cys6 transcription factors Pip2p and Oaf1p controls induction of genes encoding peroxisomal proteins in *Saccharomyces cerevisiae*. *European Journal of Biochemistry*, *247*, 776–783.
37. Saerens, S., Verstrepen, K., Van Laere, S., Voet, A., Van Dijck, P., Delvaux, F., & Thevelein, J. (2006). The *Saccharomyces cerevisiae* EHT1 and EEB1 genes encode novel enzymes with medium-chain fatty acid ethyl ester synthesis and hydrolysis capacity. *The Journal of Biological Chemistry*, *281*, 4446–4456.
38. Schadt, E. E., Monks, S. A., Drake, T., Lusis, A. J., Che, N., Colinayo, V., Ruff, T. G., Milligan, S. B., Lamb, J. R., Cavet, G., Linsley, P. S., Mao, M., Stoughton, R. B., & Friend, S. H. (2003). Genetics of gene expression surveyed in maize, mouse and man. *Nature*, *422*, 297–302.
39. Schreve, J., Sin, J., & Garrett, J. (1998). The *Saccharomyces cerevisiae* YCC5 (YCL025c) gene encodes an amino acid permease, Agp1, which transports asparagine and glutamine. *Journal of Bacteriology*, *180*, 2556–2559.
40. Stranger, B. E., Forrest, M. S., Clark, A. G., Minichiello, M. J., Deutsch, S., Lyle, R., Hunt, S., Kahl, B., Antonarakis, S. E., Tavare, S., Deloukas, P., & Dermitzakis, E. T. (2005). Genome-wide associations of gene expression variation in humans. *PLoS Genetics*, *1*(6), e78.
41. Sun, W., Yu, T., & Li, K. C. (2007). Detection of eQTL modules mediated by activity levels of transcription factors. *Bioinformatics*, *23*, 2290–2297.
42. Sun, W., Yuan, S., & Li, K. C. (2008). Trait-trait dynamic interaction: 2D-trait eQTL mapping for genetic variation study. *BMC Genomics*, *9*, 242.
43. Tenreiro, S., Nunes, P., Viegas, C., Neves, M., Teixeira, M., Cabral, M., & Sá-Correia, I. (2002). AQR1 gene (ORF YNL065w) encodes a plasma membrane transporter of the major facilitator superfamily that confers resistance to short-chain monocarboxylic acids and quinidine in *Saccharomyces cerevisiae*. *Biochemical and Biophysical Research Communications*, *292*, 741–748.
44. Troyanskaya, O., Cantor, M., Sherlock, G., Brown, P., Hastie, T., Tibshirani, R., Botstein, D., & Altman, R. B. (2001). Missing value estimation methods for DNA microarrays. *Bioinformatics*, *17*, 520–525.
45. Velasco, I., Tenreiro, S., Calderon, I., & André, B. (2004). *Saccharomyces cerevisiae* Aqr1 is an internal-membrane transporter involved in excretion of amino acids. *Eukaryotic Cell*, *3*, 1492–1503.
46. Wang, S., Yehya, N., Schadt, E. E., Wang, H., Drake, T. A., & Lusis, A. J. (2006). Genetic and genomic analysis of a fat mass trait with complex inheritance reveals marked sex specificity. *PLoS Genetics*, *2*(2), e15.
47. Wang, Y., & Dohlman, H. (2004). Pheromone signaling mechanisms in yeast: A prototypical sex machine. *Science*, *306*, 1508–1509.

48. Wu, C., Delano, D. L., Mitro, N., Su, S. V., Janes, J., McClurg, P., Batalov, S., Welch, G. L., Zhang, J., Orth, A. P., Walker, J. R., Glynne, R. J., Cooke, M. P., Takahashi, J. S., Shimomura, K., Kohsaka, A., Bass, J., Saez, E., Wiltshrie, T., & Su, A. I. (2008). Gene set enrichment in eQTL identifies novel annotations and pathway regulators. *PLoS Genetics*, *4*(5), e1000,070.
49. Yvert, G., Brem, R. B., Whittle, J., Akey, J. M., Foss, E., Smith, E. N., Mackelprang, R., & Kruglyak, L. (2003). *Trans*-acting regulatory variation in *Saccharomyces cerevisiae* and the role of transcription factors. *Nature Genetics*, *35*, 57–64.

Chapter 14
Statistical Analysis of Time Course Microarray Data

Lingyan Ruan and Ming Yuan

Abstract Time course gene expression experiments have proved valuable in a variety of studies. Their unique data structure and the diversity of tasks often associated with them present new challenges to statistical analysis. In this report, we give a brief review of several primary questions pertaining to such experiments and popular statistical tools to address them.

14.1 Introduction

Among the first microarray experiments were those measuring expression over time, and time course microarray experiments remain common. Nowadays time course data account for more than one third of microarray experiments (National Center for Biotechnology Information http://www.ncbi.nlm.nih.gov/geo/). Instead of taking a static snapshot, time course microarrays capture the dynamics of biological processes, and therefore offer a powerful tool to many biological and medical studies.

Broadly speaking, there are two primary goals in time course gene expression study. One is to characterize temporal patterns of gene expression within a single biological condition and group genes by these patterns. Doing so could provide insight into the biological function of genes if one assumes that genes with similar temporal patterns of expression share similar functions. The other goal common to many time course experiments is to collect profiles in multiple biological conditions and identify temporal patterns of differential expression. Both goals could be addressed naïvely by applying any of the many methods for analyzing static microarray data (see, e.g., [13]). For example, in a single biological condition, one could consider differential expression patterns across time points; in multiple biological conditions, each time point could be treated in isolation. However, such naïve

L. Ruan and M. Yuan (✉)
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, 755 Ferst Dr NW, Atlanta, GA 30332-0205
e-mail: lruan@isye.gatech.edu, myuan@isye.gatech.edu

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_14,
© Springer-Verlag Berlin Heidelberg 2011

approaches can be inefficient as they do not utilize the information contained in the dependence structure of the time course data. This problem is exacerbated in microarray studies, where low sensitivity is a problematic feature of many methods. In addition, a gene's expression pattern over time might not be identified by simply combining results from repeated marginal analyses.

The unique data structure of the time course data and the diversity of tasks often associated with them present new challenges to statistical analysis. One of the key issue is to model the temporal dependence among gene expression measurements. This task is often complicated by the difference in sampling schemes. In many experiments, for example, genetically identical subjects are raised and sacrificed at different time points. Therefore, the dependence across time points are primarily produced by the temporal nature of the biological process of interest. In contrast, in other experiments, the same subjects are followed up over time to collect data at different time points. The dependence between data from different time points is then determined jointly by the temporal pattern of the biological process and the subject effect. Another factor that makes the use of traditional methods difficulty is the fact that most of the time course studies involve only short time series, with less than, say, eight distinct time points.

To overcome these challenges, a number of methods have been introduced in past several years. In what follows, we shall review some of these recent advances.

14.2 Clustering of Temporal Patterns

With a large number of genes monitored, clustering is one of the foremost tasks for microarray data analysis. It identifies groups of genes that have similar expression profiles across samples. Clustering can reduce the effort of studying individual genes and more importantly it can unmask the functional groups among genes. Since the seminal work by Eisen et al. [5], various techniques have been developed for this purpose in the context of microarray experiments, which we shall roughly classify into distance-based unsupervised clustering methods and mixture model based methods.

14.2.1 Data Structure

Before reviewing some of the common ideas for clustering gene expression patterns, we first briefly discuss the data structure common to time course gene expression data where there are multiple time points; and for each time point, there are microarray measurements from possibly multiple replicates. Intensity values are background corrected and normalized to account for known sources of variation, leaving a single summary score of expression for each gene and each replicate at each time.

Table 14.1 Data structure for microarray experiments with one biological group

	Time 1				Time 2				...	Time T		
	1	2	...	n_1	1	2	...	n_2	...	1	...	n_T
Gene 1	x_{111}	x_{112}	...	x_{11n_1}	x_{121}	x_{122}	...	x_{12n_2}	...	x_{1T1}	...	x_{1Tn_T}
Gene 2	x_{211}	x_{212}	...	x_{21n_1}	x_{221}	x_{222}	...	x_{22n_2}	...	x_{2T1}	...	x_{2Tn_T}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
Gene G	x_{G11}	x_{G12}	...	x_{G1n_1}	x_{G21}	x_{G22}	...	x_{G2n_2}	...	x_{GT1}	...	x_{GTn_T}

The full observed gene expressions can be represented by a matrix whose rows correspond to genes and columns to replicates. The typical data layout is shown in Table 14.1. We denote by x_{gtk} the observed expression level for gene g on the kth array at time point t $(k = 1, \dots, n_t)$, where n_t is the number of arrays at time point t. Denote by $\mathbf{x}_{gt} = (x_{gt1}, \dots, x_{gtn_t})$ the row vector of gene expressions for gene g at time t, and by $\mathbf{x}_g = (\mathbf{x}_{g1}, \dots, \mathbf{x}_{gT})$ all gene expressions collected for gene g.

14.2.2 Distance-based Clustering

Unsupervised clustering of gene expressions is generally based upon a certain distance measure reflecting the similarity or dissimilarity of the temporal expression patterns between genes. For example, Eisen et al. [5] propose to quantify the similarity between two temporal expression patterns by their Peason's correlation coefficient, i.e., $\text{Corr}(\log(\mathbf{x}_g), \log(\mathbf{x}_{g'}))$ between genes g and g'. In a similar spirit, Tavazoie et al. [21] suggested to use the Euclidean distance between two expression profiles, i.e.,

$$\| \log(\mathbf{x}_g) - \log(\mathbf{x}_{g'}) \| = \left(\sum_{t,k} \left(\log(x_{gtk}) - \log(x_{g'tk}) \right)^2 \right)^{1/2}.$$

Note that correlation coefficients are similarity measure in that the bigger the score, the more similar the two expression profiles; whereas Euclidean distance is a dissimilarity measure in that the smaller the distance, the more similar the two profiles. Both of them can be used for the purpose of unsupervised clustering. Other similarity or dissimilarity measures have also been introduced to take the temporal nature of the expression data into account (see, e.g., [6,17]). One notable example is Spellman et al. [17] who studied the cyclic patterns of time course data collected in yeast. They propose to measure the similarity between temporal expressions in the Fourier domain, which allows for the identification of the peak expression during the cell cycle.

Once the pairwise distance is constructed for the genes, various unsupervised clustering techniques can be applied. One of the popular choice is hierarchical clustering which, for example, was adopted by Eisen et al. [5]. The hierarchical clustering computes a dendrogram that assembles all elements into a single tree. In a nutshell, hierarchical clustering proceeds as follows. Let S be the similarity matrix computed using any of the aforementioned methods, i.e., $S_{gg'}$ is the numerical score representing the similarity between genes g and g'. The largest entry in the matrix is first identified and the corresponding genes are assigned to the same cluster because they have the greatest similarity. The average gene expressions of the two genes can be then be used in place of the two individual ones to represent the cluster. The similarity matrix is then updated with the similarity between this new cluster and remaining genes computed. The process then continues by screening the greatest similarity and combining the corresponding genes or clusters. The procedure stops until all genes are merged into a single cluster the relationship among genes is now represented by a tree structure.

Another popular similarity based clustering approach is the k-means algorithm which has been used by Tavazoie et al. (1999) among others in the context of time course microarray data. It iteratetively minimizes the within-cluster sum of squared distances from the cluster mean. The first cluster centre is the centroid of the entire data set and subsequent centers are decided by finding the data point farthest from the centers already chosen. The algorithm is repeated until convergence that the cluster memberships do not change appreciably between iterations.

Self-organizing maps (SOMs) have also been used to group temporal patterns of gene expressions [20]. It has a set of nodes with a simple topology and a distance function on the nodes. Nodes are iteratively mapped into a gene expression space. The initial mapping is random and the subsequent iterations re-position the nodes by moving toward a selected point, which is chosen based on random ordering. The movement depends on the initial geometry with the closet nodes moving the most. Neighboring points are identified as clusters and a structure in data is imposed by SOMs.

14.2.3 Model-Based Clustering

In addition to the distance based clustering techniques, model based clustering methods have also been developed. In these approaches, each cluster is represented by a component in a mixture model. For a moment, suppose we know apriori that there are C clusters among the genes. Expression measurements for genes from the same cluster are expected to have similar profiles, and therefore can be reasonably modeled as observations from the same distribution. More specifically, if gene g comes from the kth cluster, then $\mathbf{x}_g \sim f_k(\mathbf{x}_g)$. Under this notion, measurements of a randomly picked gene g from the G genes we observed should follow

$$\mathbf{x}_g \sim \pi_1 f_1(\mathbf{x}_g) + \ldots + \pi_C f_C(\mathbf{x}_g)$$

where π_k is the prior probability that g comes from the kth cluster ($\pi_1 + \ldots + \pi_C = 1$).

Different choices of the component distribution f_k have been researched in the literature. Ramoni, Sebastiani, and Kohane [15] consider each gene's expression profile as output from an autoregressive (AR) process. Schliep, Schönhuth, and Steinhoff [16] address similar goals. In their work, partially supervised learning is used to identify an initial set of clusters at each time point, represented by a hidden Markov model (HMM). Along the same vein, Bar-Joseph et al. [1] and Luan and Li [11] proposed to model temporal patterns of gene expressions from a cluster by splines with splines coefficients coming from a common distribution.

Once the component distribution is known, clustering can be done based upon posterior calculations. For example, the chance that a gene comes from cluster k can be conveniently represented by the following posterior probability:

$$P\left(\text{gene } g \text{ comes from cluster } k|\mathbf{x}_g\right) = \frac{\pi_k f_k(\mathbf{x}_g)}{\pi_1 f_1(\mathbf{x}_g) + \ldots + \pi_C f_C(\mathbf{x}_g)}.$$

In principal, estimation of the parameters involved in the component distributions can be done through maximum likelihood, which is often implemented using variants of EM algorithm. In practice, however, it can be quite challenging computationally. Heuristic algorithms are often employed to alleviate such problems. Another difficulty with model-based approaches is in the choice of the number of clusters, C, which amounts to a model selection problem. BIC criterion is often used for such purpose although other heuristic methods are also popular in practice.

More recently, Wu et al. [24] introduced a partially Bayesian hierarchical model integrated with a hidden Markov Model (HMM) structure and auto-regression (AR) to model temporal gene expression profiles at both the expression level and the state level to identify genes with trajectories that change over time and to group genes with similar trajectories.

14.2.4 Related Issues

Depending on the context of the study, several recurring issues often arise with clustering temporal gene expression data. For example, periodic phenomenon is common in a lot of studies. An efficient and interpretable clustering procedure should therefore takes such behavior into account (see, e.g., [22]; Kim et al. [2]). In particular, Kim et al. (2006) propose to approximate periodical expression levels by Fourier Series of order one. Distance based clustering can then be employed with similarities or dissimilarities among the genes measured in terms of the phase and amplitude of the Fourier series.

In a similar spirit, one may consider project the dynamic profile of a gene's expression level into a small number of 'characteristic' modes and then cluster genes according to the projection. One of the more popular examples is the singular value decomposition (SVD) approach from Holter et al. (2000) where the first few singular vectors are used as the basis expression patterns.

Another task that is somewhat related to clustering is the identification of genes that show differential expression over time. To this end, let $\mu_{gt} = \mathbb{E}(\log(x_{gtk}))$ be the average log expression level for gene g at time t. The problem can be naturally formulated as classifying a gene into two possible clusters. Majority of the genes fall into the first class where

$$\mu_{g1} = \mu_{g2} = \cdots = \mu_{gT}, \tag{14.1}$$

whereas others are classified into a class where this relationship does not hold. In other words, the second cluster of genes demonstrates differential expression over time and this could be due to biological response to a certain treatment or triggering event at the beginning of the experiment. This problem can be naturally cast as a hypothesis testing problem and treated in a similar fashion as the differential expression identification for static gene expression analysis (see, e.g., [19]).

14.2.5 Data Example

To demonstrate the usefulness of clustering in time course gene expression studies, we revisit an experiment on memory CD8 T cell differentiation. The experiment, original reported in Kaech et al. [10], was done in the context of a large research effort to understand immune memory in Rafi Ahmeds laboratory of the Emory Vaccine Research Center. Here immune memory refers to the ability of the immune system to remember its rst exposure to a specic antigen and to mount a rapid and aggressive response to a second exposure. In the immune system, CD8 T cells are specialized immune cells that play an important role in the regulation of antiviral response and the generation of protective immunity. In response to a viral infection, naïve CD8 T cells differentiate into effector CD8 T cells that control the infection and the effector CD8 T cells that survive continue to differentiate into long-lived protective memory CD8 T cells.

In this particular experiment, acute lymphocytic choriomeningitis virus Armstrong (LCMV) infection of mice was used as a model system to study CD8 T cell development. Genetically identical, uninfected mice were sacrificed on the baseline day to obtain naïve CD8 T cells. Other genetically identical mice were infected with LCMV on the baseline day. Then mice were sacrificed at day 8, day 15, and greater than day 30 (Imm). Because mice were sacrificed at each time point, each time point contributes an independent sample. Cells from several mice were pooled for each microarray chip to increase stability. Further detailed information about this experiment can be found in Kaech et al. (2002).

In the original analysis, Kaech et al. (2002) selected genes based on whether their average gene expression changed (decreased or increased) by at least 1.7 fold between any two time points, generating a set of 431 genes. After that, they applied a K-means clustering algorithm on these genes and found 6 major patterns. Although the results are useful to a certain extent, the temporal aspects of the data

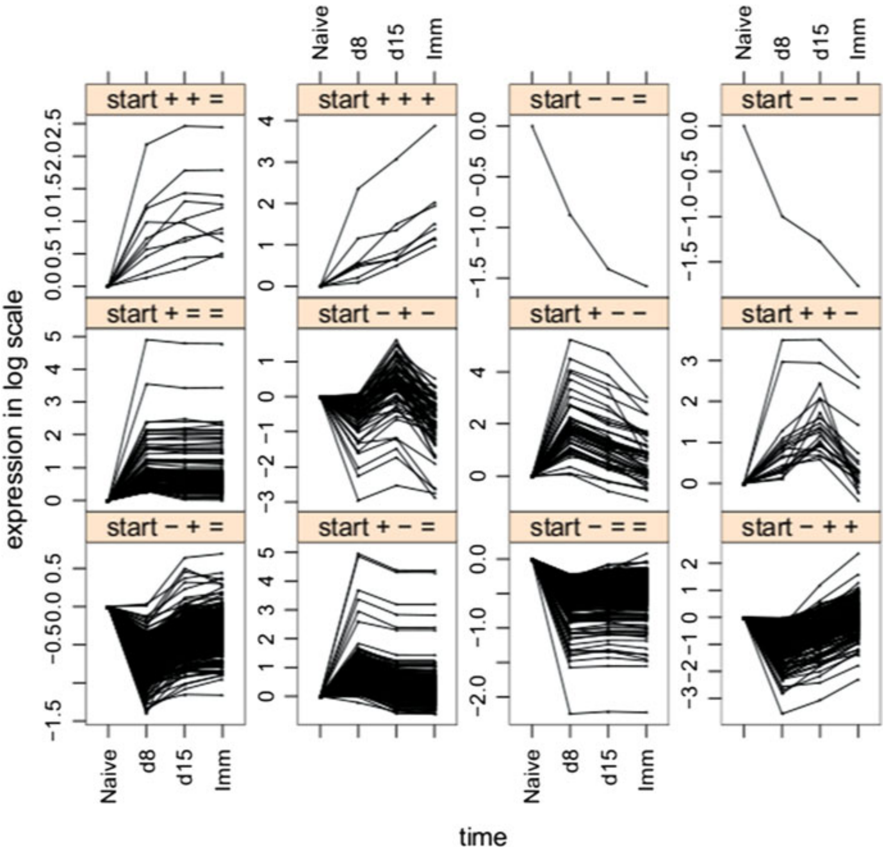


Fig. 14.1 Model based clustering for the immune data

were ignored in this analysis. Moreover, both the fold change cutoff in the selection method and the number of clusters in K-means clustering were chosen arbitrarily. As a result, the biological meaning of the obtained clusters is not clear and the interpretation of clustering results is not straightforward.

To improve interpretation, Wu et al. [24] reanalyzed this CD8 T-cell experiment by focusing on the direction (upregulation, downregulation and no change) and the magnitude of the gene-specic successive differences (changes) in the mean gene expression levels (log base 2 scale) over time. The clustering result is reproduced in Fig. 14.1 where upregulation, downregulation and no change between two successive time points are represented by +, − and = respectively.

In particular, when compared with the results obtained from K-means clustering (Kaech et al., 2002), both methods identify clusters ‘start,+,−,−’, ‘start,−,+,+’, and ‘+,=,=’ as the more prominent clusters and often associated with important biomarkers:

Cluster ‘start, $+$, $-$, $-$’: Upregulated at day 8 and gradually downregulated at day 15 and memory stage. This cluster contains 46 genes. Important genes in this cluster include GZMA, GZMB, GZMK KLRG1, and CCR2.

Cluster ‘start, $-$, $+$, $+$’: Downregulated at day 8 and gradually upregulated at day 15 and memory stage. This cluster contains 192 genes. Important genes in this cluster contains genes like IL7R, BCL2, CXCR4, CD62L, and CCR7.

Cluster ‘start, $+$, $=$, $=$’: Upregulated at day 8 with no change over the following time points. This cluster contains 282 genes. Important genes in this cluster include KLRA3, CCL9 and CD44.

14.3 Temporal Differential Expression

When gene expression measurements come from multiple biological conditions, a fundamental goal is to identify those genes which are differentially expressed under different conditions. This practice oftentimes helps investigators identify specific diagnostic, prognostic and predictive factors for disease which can ultimately lead to the development of molecular-based therapies. The development of statistical methods to identify differentially expressed genes for static microarray data has received much attention, especially methods to identify genes that are differentially expressed between two conditions. For detailed discussion regarding this subject, the readers are referred to Parmigiani et al. [13] and the references therein.

The general data structure are similar to before except that the replicates are now collected under at least two biological conditions. The primary goals of the study are to identify genes with different levels of expression over time. Following the previous notation, we shall write x_{gtkc} in what follows with an additional subscript c to denote the conditions, i.e., for each c, $\{x_{gtkc} : g, t, k\}$ would have the same layout as given in Table 14.1.

In contrast to the rich literature on static microarray experiments, fewer papers are dedicated to time course microarray experiments. Methods up to now for microarray time course experiments can be roughly divided into three classes: (1) methods extended from those for static microarray experiments; (2) methods based on smoothing; and (3) methods extended from time series analysis.

14.3.1 ANOVA

Expression measurements for a gene can be regarded as noisy observations of a vector of latent expression levels at different time and for different biological conditions. In the current setup, we use μ_{gtc} to represent the latent level for the log-transform expression of gene g at time t under condition c, i.e.,

$$\log(x_{gtkc}) = \mu_{gtc} + \epsilon_{gtkc}.$$

Under the independent sampling when each array is done on independent subjects, the measurement errors ϵ_{gtkc} can be treated as independent. In contrast, when longitudinal sampling scheme was adopted such that the kth replicate is always measured on the same subject, the measurement errors $\{\epsilon_{gtkc} : t = 1, \ldots, T\}$ are correlated. Various models for the correlation structure for the measurement errors have been studied in details in the literature of longitudinal data analysis (see, e.g., [3]).

With this setup, gene g is equivalent expressed if and only if $\mu_{gt1} = \mu_{gt2} = \ldots$ for all $t = 1, \ldots, T$. Various test statistics (see, e.g., [19]) have been introduced in recent year to test this hypothesis. One popular class of such method is based upon ANOVA where the following models are often considered (see, e.g., [12]):

$$\mathscr{M}_1 : \quad \log(x_{gtkc}) = \mu_g + \alpha_{gc} + \beta_{gt} + \epsilon_{gtkc};$$

and

$$\mathscr{M}_2 : \quad \log(x_{gtkc}) = \mu_g + \alpha_{gc} + \beta_{gt} + \gamma_{gtc} + \epsilon_{gtkc},$$

where μ_g models the gene effect, α_{gc} and β_{gt} represent the gene-specific condition effect and temporal effect, and γ_{gtc} can be used to incorporate the two-way interaction between time and condition.

Based on these models, if gene g is equivalently expressed, then $\alpha_{gc} = 0$ and $\gamma_{gtc} = 0$. This can be cast as a typical hypothesis testing problem and tested using F-test when assuming that the idiosyncratic noise ϵ_{gtkc} follows a centered normal distribution. The normality assumption can be relaxed using permutation test. Adjustment to multiple comparisons can also be done in a similar fashion as the static case. The main drawback of this class of method is that they do not account for the temporal order of the time course data, i.e., identical results would be obtained if the time points were reordered. As a result, they are susceptible to low sensitivity.

14.3.2 Smoothing

To further account for the temporal nature of time course data, smoothing based methods have also been introduced. It is now assumed that $\mu_{gtc} = f_{gc}(t)$ where f_{gc} is a smooth function. In this setting, identifying temporal differential expression can be formulated as testing $H_0 : f_{g1} = f_{g2} = \ldots$. The main challenge is how to model the true gene expression profile described by the function $f_{gc}(\cdot)$. Hong and Li (2006) and Storey et al. [18], among others, propose to model it as a linear combination of a finite set of B-spline basis functions. The rationale behind this modelling approach is the assumption that the temporal process evolves possibly nonlinearly but smoothly and this smoothness is governed by the number of basis functions used in modelling $f_{gc}(\cdot)$. How well the profile can be approximated heavily depends on the number of basis functions and their respective locations. Unfortunately, the selection of basis functions, most of the time, can only be done on a case-to-case basis. A flexible alternative to the B-spline approach is to view $f_{gc}(\cdot)$ as a realization of a Gaussian process (Yuan, 2006).

The idea of Gaussian process modeling is, without parametrizing a function, to view it as a sample from the space of functions. A Gaussian process defines a distribution over functions. It can be thought of as the generalization of a multivariate normal distribution over a finite vector space to a function space of infinite dimension. Different from parametric approaches such as the one used by Hong and Li (2004) where inferences about a function is made via the inference on the linear coefficients, any inference regarding the function takes place directly in function space with Gaussian process modeling. The Gaussian process based approach of Yuan (2007) can also conveniently handle more than two biological conditions.

To elaborate on this, consider the aging experiment from Edwards et al. [4]. The experiment was designed to better understand the genetic basis underlying the relationship between longevity and the ability to resist oxidative stress as we shall discuss in detail later. After stress induction, the investigators monitored the gene expression level for young, middle-aged and old mice at five different time points. There are no natural ways of applying Hong and Li's approach to compare the three age groups. Storey et al.'s approach can tell us which genes are not equivalently expressed across all three groups. But it can not provide information on whether or not the differential expression occurs only for one group. Furthermore, the validity of both existing profile-modelling methods is questionable with such a small number of time points. In contrast, the Gaussian process approach classifies the genes according to the following five patterns.

$$\begin{aligned} H_1 &: f_{g,aged} = f_{g,middle} = f_{g,young} \\ H_2 &: f_{g,aged} \neq f_{g,middle} = f_{g,young} \\ H_3 &: f_{g,middle} \neq f_{g,aged} = f_{g,young} \\ H_4 &: f_{g,young} \neq f_{g,aged} = f_{g,middle} \\ H_5 &: f_{g,aged} \neq f_{g,middle} \neq f_{g,young}. \end{aligned}$$

Among a total of 10,043 genes, 7,396 genes to H_1 (equivalent expression); 369 genes to H_2 (Aged differentially expressed); 731 to H_3 (Middle-aged differentially expressed); 1,467 to H_4 (Young differentially expressed), and 80 to H_5 (all three conditions differentially expressed). Figure 14.2 depicts the expression measurements and estimated expression profiles for a sample of 15 genes. The black circles, red triangles and green pluses represent the expression measurements taken for aged, middle-aged and young age group respectively. The solid lines are the estimated expression profile and the broken lines stand for the 99% Bayesian confidence bands. The three genes from the first column are identified as H_1, and as indicated by the plot, the three estimated expression profiles are very similar. The second to fourth column each has three genes classified to pattern H_2, H_3 and H_4 respectively, where one age group shows different expression profile from the other two. The fifth column corresponds to pattern H_5. These genes have three different expression profiles under different conditions. Such plot not only helps us determine a gene's expression pattern but also visualizes a gene expression trajectory under different conditions.

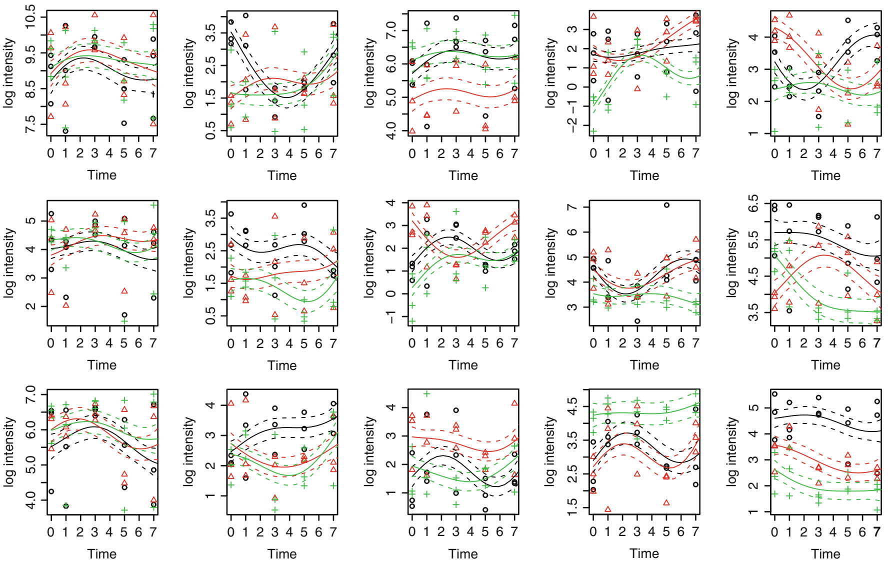


Fig. 14.2 Genes with different patterns expression

14.3.3 Differential Expression Pattern

The primary goals of many time course studies are to identify genes with different levels of expression at each time and classify genes into temporal expression patterns. The aforementioned approaches target at genes that are temporally expressed but can not address the question of when the differential expression occurs and there is no information indicating which time points contribute most to a gene being identified as DE across conditions. To address this question, Yuan and Kendziorski [26] proposed a HMM method where a gene's differential expression pattern at each time point $\{s_{gt} : t = 1, \ldots, T\}$ is modeled by an unobservable Markov Chain, s_{gt} can be either differential expression or equivalent expression. Inference about a gene's expression pattern at different time points can then be inferred based upon observed expression measurements. This method is easy to implement thanks to dynamic programming techniques such as the Baum-Welch and Viterbi algorithms. To illustrate the variety of questions that often arise with comparing multiple biological conditions and the versatility of the HMM approach, we reproduce below the analysis of a couple of case studies from Yuan et al. [27].

The first study concerns the effects of type 2 diabetes on the kidney. The obese Zucker rat is a model organism for studies of this type. Affymetrix Rat Expression set 230 chips were used to measure the expression levels of 15,923 genes in the kidney tissue of control and treated obese Zucker rats over time. The treated rats received streptozotocin, a chemical known to selectively damage islet cells in the pancreas thus increasing the progression to type 2 diabetes. Expression

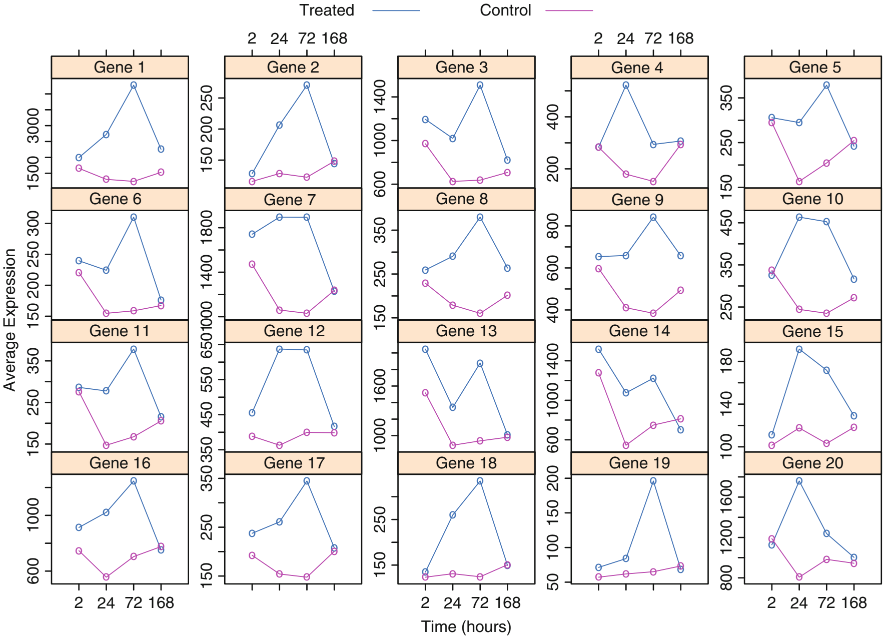


Fig. 14.3 Average expression for genes that are differentially expressed in the middle of the study

measurements were obtained at 2 h, 1 day, 3 days, and 7 days following treatment. Two rats were considered for each time and age combination for the first three time points; three rats in each combination were considered on day 7. All Affymetrix image files were processed using the *affy* software in Bioconductor [9]; intensity scores for each gene were obtained via RMA. In this study, the investigators are primarily interested in genes that show differential expression only in the middle of the time course as they are more likely to be responsive to the initial treatment. This can be expressed in terms of the expression pattern: genes that are equivalently expressed (i.e., $\mu_1 = \mu_2$ at 2 h and 7 days, but differentially expressed (i.e., $\mu_1 \neq \mu_2$) at 24 h and 72 h. Genes following such pattern can be conveniently identified using the HMM method and a subset of them are shown in Fig. 14.3

The second case study considers the genetic response to varying doses of 2,3,7,8-Tetrachlo-rodibenzo-p-dioxin (dioxin) in mouse liver tissue. Dioxin is the prototype for a family of highly toxic compounds widely dispersed in the environment. Exposure to dioxin can lead to liver damage, endocrine disruption, birth defects, and cancer [14, 23]. cDNA microarrays were used to measure the expression levels of 1,536 genes in liver tissue of mice treated with three doses of dioxin (low, medium, high). Seven time points were considered (1,2,4,8,16,32, and 64 days following dioxin). Four mice were measured for each time and dose combination. Dye swaps were also done to give a total of 168 arrays; 132 arrays provided useable data. Spotfire (www.spotfire.com) was used to process the image files and correct

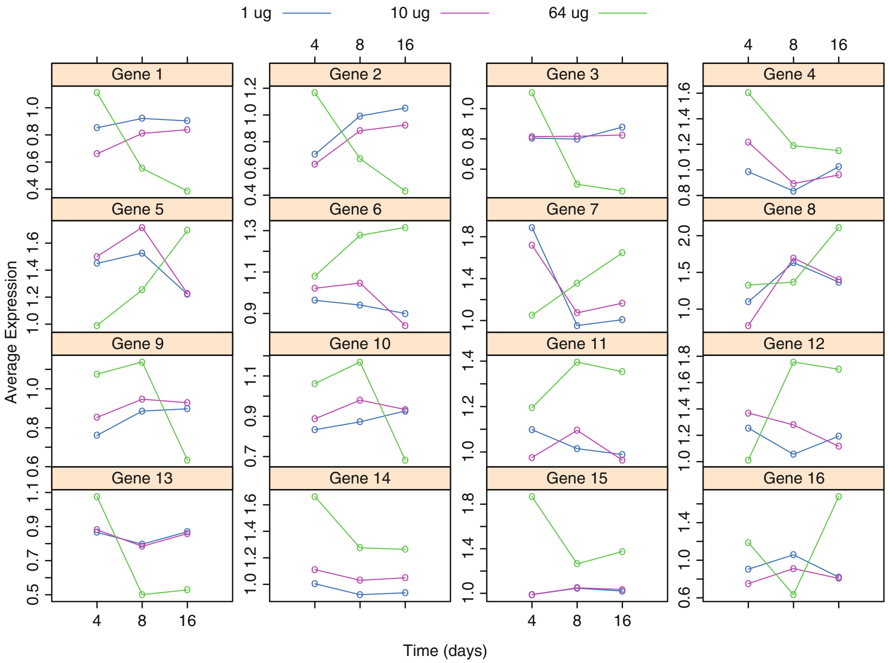


Fig. 14.4 Average expression for genes that are differentially expressed in the middle of the study

for dye effects. Of particular interest in the study is to identify genes showing equivalent expression between low doses, but differential expression at the higher dose at the intermediate time points. Viterbi paths corresponding to this pattern ($\mu_1 = \mu_2 \neq \mu_3$) at Day 4, Day 8 and Day 16 were identified; the expression profiles for a subset of these genes are shown in Fig. 14.4.

14.4 Concluding Remarks

Time course expression experiments have been successfully used to study a wide range of biological systems as they provide key information about the dynamics of complex biological systems. Driven by the need to effective extract information from these data, a growing number of approaches have been developed in recent years to address various questions associated with this type of experiments. In this report, we gave a short review of some of the representative ideas in clustering and identifying differential expression in time course studies. The review is by no means exhaustive but to demonstrate the unique nature of the data structure and multitude of questions that often arise with time course studies.

Acknowledgements This research was supported in part by NSF grants DMS-0706724 and DMS-0846234, NIH grant R01GM076274-01, and a grant from Georgia Cancer Coalition.

References

1. Bar-Joseph, Z., Gerber, G., Gifford, D. K., Jaakkola, T. S., & Simon, I. (2003). Continous representations of time-series gene expression data. *Journal of Computational Biology*, *10*, 341–356.
2. Do, K.-A., Mueller, P., & Vanucci, M. (Eds.). (2006). Bayesian inference for gene expression and proteomics. Cambridge University Press.
3. Diggle, P. J., Heagerty, P., Liang, K.-Y., & Zeger, S. L. (2002). *Analysis of longintudinal data* (2nd ed.). New York: Oxford University Press.
4. Edwards, M. G., Sarkar, D., Klopp, R., Morrow, J. D., Weindruch, R., & Prolla, T. A. (2003). Age-related impairment of the transcriptional response to oxidative stress in the mouse heart. *Physiological Genomics*, *13*, 119–127.
5. Eisen, M. B., Spellman, P. T., Brown, P. O., & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, *95*, 14863–14868.
6. Ernst, J., Nau, G. J., & Bar-Joseph, Z. (2005). Clustering short time series gene expression data. *Bioinformatics*, *21*, i159–i168.
7. Hong, F., & Li, H. (2006). Functional hierarchical models for identifying genes with different time-course expression profiles. *Biometrics*, *62*, 534–544.
8. Holter, N. S., Mitra, M., Maritan, A., Cieplak, M., Banavar, J. R., & Fedoroff, N. V. (2000). Fundamental patterns underlying gene expression profiles: Simplicity from complexity. *Proc. Natl. Acad. Sci. USA*, *97*, 8409–8414.
9. Ihaka, R., & Gentleman, R. (1996). R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics*, *5*, 299–314.
10. Kaech, S.M., Hemby, S., Kersh, E., & Ahmed, R. (2002). Molecular and functional profiling of memory CD8 T cell differentiation. *Cell*, *111*, 837–851.
11. Luan, Y., & Li, H. (2003). Clustering of time-course gene expression data using a mixed-effects model with B-splines. *Bioinformatics*, *19*(4), 474–482.
12. Park, T., Yi, S. G., & Lee, S. (2003). Statistical tests for identifying differentially expressed genes in time-course microarray experiments. *Bioinformatics*, *19*, 694–703.
13. Parmigiani, G., Garrett, E., Irizarry, R., & Zeger, S. (2003). *The Analysis of gene expression data: Methods and software*. New York: Springer Verlag.
14. Poland, A., & Knutson, J. C. (1982). 2,3,7,8-tetrachlorodibenzo-p-dioxin and related halogenated aromatic hydrocarbons: Examination of the mechanism of toxicity. *Annual Review of Pharmacology and Toxicology*, *22*, 517–554.
15. Ramoni, M. F., Sebastiani, P., & Kohane, I. S. (2002). Cluster analysis of gene expression dynamics. *Proceedings of the National Academy of Sciences*, *99*, 9121–9126.
16. Schliep, A., Steinhoff, C., & Schönhuth, A. (2003). Robust inference of groups in gene expression time-courses using mixtures of HMMs. *Bioinformatics*, *20*, i283–i289.
17. Spellman, P. T., Sherlock, G., Zhang, M., Iyer, V. R., Anders, K., Eisen, M. B., et al. (1998). Comprehensive identification of cell-cycle regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, *9*, 3273–3297.
18. Storey, J. D., Xuai, W., Leek, J. T., Tompkins, R. G., & Davis, R. W. (2005). Significance analysis of time course microarray experiments. *Proceedings of the National Academy of Sciences of the United States of America*, *102*, 12837–12842.
19. Tai, Y. C., & Speed, T. P. (2006). A multivariate empirical Bayes statistic for replicated microarray time course data. *Annals of Statistics*, *34*, 2387–2412.

20. Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., et al. (1999). Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences of the United States of America*, *96*, 2907–2912.
21. Tavazoie, S., Hughes, J., Campbell, M., Cho, R., & Church, G. (1999). Systematic determination of genetic network architecture. *Nature Genetics*, *22*, 281–285.
22. Tu, B., Kudlicki, A., Rowicka, M., & McKnight, S. (2005). Logic of the yeast metabolic cycle: Temporal compartmentalization of cellular processes. *Science*, *310*, 1152–1158.
23. Wilson, C. L., & Safe, S. (1998). Mechanisms of ligand-induced aryl hydrocarbon receptor mediated biochemical and toxic responses. *Toxicologic Pathology*, *26*, 657–671.
24. Wu, H., Yuan, M., Kaech, S., & Halloran, M. (2007). A statistical analysis of memory CD8 T cell differentiation: An application of a hierarchical state space model to a short time course microarray experiment. *Annals of Applied Statistics*, *1*(2), 442–458.
25. Yuan, M. (2006). Flexible temporal expression profile modelling using the Gaussian process. *Computational Statistics and Data Analysis*, *51*, 1754–1764.
26. Yuan, M., & Kendziorski, C. (2006). Hidden markov modles for microarray time course data in multiple biological conditions (with discussion). *Journal of the American Statistical Association*, *101*, 1323–1340.
27. Yuan, M., Kendziorski, C., Park, F., Porter, J., Hayes, K., & Bradfield, C. (2003). *HMM for microarray time course data in multiple biological conditions* (Technical Report # 178). Department of Biostatistics, University of Wisconsin.

Part III
Systems Biology

Chapter 15
Kernel Methods in Bioinformatics

Karsten M. Borgwardt

Abstract Kernel methods have now witnessed more than a decade of increasing popularity in the bioinformatics community. In this article, we will compactly review this development, examining the areas in which kernel methods have contributed to computational biology and describing the reasons for their success.

15.1 Introduction

Kernel methods are a family of algorithms from statistical machine learning [61,67]. These include the Support Vector Machine (SVM) for regression and classification as well as methods for principal component analysis [62], feature selection [72], clustering [94], two-sample tests [7, 19], or dimensionality reduction [93]. These kernel methods have witnessed a huge surge in popularity in bioinformatics over the last decade. To illustrate this popularity: `pubmed`, the search engine for biomedical literature, lists 1,710 hits for 'kernel methods' and 1,798 hits for 'SVM' (as of May 28, 2009).

The goal of this article is to review which problems in bioinformatics have been tackled using kernel methods, and to explain their popularity in this field. Section 15.2 provides a summary of the central terminology in kernel methods. Section 15.3 describes how kernels can be used for data integration. Section 15.4 illustrates the power of kernel methods in dealing with structured objects such as strings or graphs. Section 15.5 presents an overview of applications of Support Vector Machines in bioinformatics, and Sect. 15.6 reviews applications of kernel methods in bioinformatics beyond SVM-based classification or regression. The interested reader is referred to Chaps. 10 and 2 of Schölkopf et al. [63] for primers on molecular biology and kernel methods, to an introduction to Support Vector

K.M. Borgwardt
Machine Learning and Computational Biology Research Group, Max Planck Institute for Intelligent Systems and Max Planck Institute for Developmental Biology, Tübingen, Spemannstr. 38, Tübingen, Germany
e-mail: karsten.borgwardt@tuebingen.mpg.de

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_15,
© Springer-Verlag Berlin Heidelberg 2011

Machines and kernel methods in computational biology [4], and to a primer on Support Vector Machines for biologists [49].

15.2 Terminology

A *kernel function* is an inner product between two objects $x, x' \in \mathscr{X}$ in a feature space $\mathscr{H}$:

$$k(x, x') = \langle \phi(x), \phi(x') \rangle \tag{15.1}$$

where $\phi : \mathscr{X} \rightarrow \mathscr{H}$ maps the data points from the input space X to feature space $\mathscr{H}$. $k(x, x')$ is referred to as the *kernel value* of x and x'. If this kernel function is applied to all pairs of objects from a set of objects, one obtains a matrix of kernel values, the *kernel matrix* K. K is always positive semi-definite,* that is all its eigenvalues are non-negative. Intuitively, a kernel function can be thought of as a similarity function between x and x', and $k(x, x')$ can be thought of as a similarity score, and the matrix K as a similarity matrix, that is a matrix of similarity scores.

The idea underlying kernel methods is to map the original input data, on which statistical inference is to be performed, to a higher dimensional space, the so-called *feature space*, and to perform inference in this feature space. Naively, this procedure would comprise two steps: (1) mapping the data points to feature space via a mapping ϕ, (2) performing the prediction or computing the statistics of interest in this feature space. Kernel methods manage to perform this procedure in one single step: rather than separating mapping and prediction into two steps, inference is performed by evaluating kernel functions on the objects in input space. By means of these kernel functions, one implicitly solves the problem in feature space, but without explicitly computing the mapping ϕ. Hence any algorithm that solves a learning problem by accessing the data points only by means of kernel functions is a *kernel method*.

15.3 Data Integration

One major reason for the popularity of kernel methods in bioinformatics is their power in data integration. This attractiveness is due to the closure properties which kernels possess:

1. k_1, k_2 are kernels $\Rightarrow k = k_1 + k_2$ is a kernel
2. k_1, k_2 are kernels $\Rightarrow k = k_1 * k_2$ is a kernel
3. k_1 is a kernel, λ is a positive scalar $\Rightarrow k = \lambda * k_1$ is a kernel

* The machine learning community often (incorrectly) uses the term *positive definite* rather than *positive semi-definite*.

Hence kernels can easily be combined in linear combinations or products. For instance, to compare two proteins, one can define a kernel on their sequences and on their 3D structures and then combine these into a joint sequence–structure kernel for proteins [40].

The goal of *multiple kernel learning* is to optimise the weights in a linear combination of kernels for a particular prediction task [34]; a related technique is referred to as *hyperkernels* [50]. Lack of runtime efficiency turned out to be a limitation of early approaches to multiple kernel learning and triggered further research that addressed this problem [54, 75]. In bioinformatics, [35] applied the kernel learning technique to protein function prediction by optimally combining kernels on genome-wide data sets, including amino acid sequences, hydropathy profiles, gene expression data and known protein-protein interactions. Tsuda et al. [84] present an efficient variant of multiple kernel learning for protein function prediction from multiple networks, such as physical interaction networks and metabolic networks.

15.4 Analysing Structured Data

A second advantage of kernel methods is that they can easily be applied to structured data [22], for instance, graphs, sets, time series, and strings. The single requirement is that one can define a positive definite kernel on two structured objects, which intuitively speaking, quantifies the similarity between these two objects. As strings are abundant in bioinformatics as nucleotide and amino acid sequences, and biological networks steadily gain more attention, this applicability to structured data is another reason for the popularity of kernel methods in bioinformatics. In the following, we will describe the basic concepts underlying string and graph kernels.

15.4.1 String Kernels

The classic kernel for measuring the similarity of two strings s and s' from an alphabet Σ is the *spectrum kernel* [36] that counts common substrings of length n in the two strings:

$$k(s, s') = \sum_{q \in \Sigma^n} \#(q \subseteq s)\#(q \subseteq s'), \tag{15.2}$$

where $\#(q \subseteq s)$ is the frequency of substring q in string s, which can be computed in $O(|s| + |s'|)$ [89], where $|s|$ is the length of string s.

As nucleotide and protein sequences are prone to mutations, insertions, deletions and other changes over time, the spectrum kernel was extended in several ways to allow for mismatches [37], substitutions, gaps and wildcards [38]. Recently, the runtime of these string kernels with inexact matching was sped up significantly in

Kuksa et al. [32]. Approaches such as [74] allow to perform SVM training on very large string datasets.

15.4.2 Graph Kernels

The classic kernel for quantifying the similarity of two graphs is the random-walk graph kernel [17, 28], which counts matching walks in two graphs. It can be computed elegantly by means of the direct product graph, also referred to as tensor or categorical product [26].

Definition 15.1. Let $G = (V, E, \mathscr{L})$ be a graph with vertex set V, edge set E and a label function: $\mathscr{L} : V \cup E \to \mathbb{R}$. The direct product of two graphs $G = (V, E, \mathscr{L})$ and $G' = (V', E', \mathscr{L}')$ shall be denoted as $G_\times = G \times G'$. The node set $V_\times$ and edge set $E_\times$ of the direct product graph are defined as:

$$\begin{aligned} V_\times &= \{(v_i, v'_{i'}) : v_i \in V \wedge v'_{i'} \in V' \wedge \mathscr{L}(v_i) = \mathscr{L}'(v_{i'})\} \\ E_\times &= \{((v_i, v'_{i'}), (v_j, v'_{j'})) \in V_\times \times V_\times : \\ &\qquad (v_i, v_j) \in E \wedge (v'_{i'}, v'_{j'}) \in E' \wedge (\mathscr{L}(v_i, v_j) = \mathscr{L}'(v'_{i'}, v'_{j'}))\} \end{aligned} \tag{15.3}$$

Using this product graph, the random walk kernel (also known as *product graph kernel*) can be defined as follows.

Definition 15.2. Let G and G' be two graphs, let $A_\times$ denote the adjacency matrix of their product graph $G_\times$, and let $V_\times$ denote the node set of the product graph $G_\times$. With a sequence of weights $\lambda = \lambda_0, \lambda_1, \ldots (\lambda_i \in \mathbb{R}; \lambda_i \geq 0 \text{ for all } i \in \mathbb{N})$ the product graph kernel is defined as

$$k_\times(G, G') = \sum_{i,j=1}^{|V_\times|} [\sum_{k=0}^{\infty} \lambda_k A_\times^k]_{ij} \tag{15.4}$$

if the limit exists.

Naively implemented, random walk kernels scale as $O(n^6)$, where n is the number of nodes in the larger of the two graphs, but their runtime was reduced to $O(n^3)$ by means of Sylvester equations [91]. As random walk kernels are limited in their ability to detect common (non-path-shaped) substructures, a family of graph kernels has been proposed that count other types of matching subgraph patterns, for instance shortest paths [8], cycles [24], subtrees [55], and limited-size subgraphs [69].

In recent work [68], a highly-scalable graph kernel was presented based on so-called *subtree patterns* or *tree-walks*. Its runtime scales as $O(N\ h\ m)$, where N is the number of graphs in the dataset, h the height of the subtree patterns and m the number of edges per graph. This graph kernel is orders of magnitude faster than previous approaches, while leading to competitive or better results on several benchmark datasets.

15.5 Support Vector Machines in Bioinformatics

The ultimate reason why kernel methods became a central branch of statistical bioinformatics was the Support Vector Machine, which reached or outperformed the accuracy levels of state-of-the-art classifiers on numerous prediction tasks in computational biology. For a comprehensive review of Support Vector Machines in computational biology up to the year 2004, the interested reader is referred to Noble [48].

Support Vector Machines were originally defined for binary classification problems [11, 85]: Given two classes of data points, a positive and a negative class, one wants to be able to correctly predict the class membership of new, unlabeled data points. Support Vector Machines tackle this task by introducing a hyperplane that separates the positive from the negative class, and which maximises the margin, that is the distance to any point from the positive or negative class. New data points are then predicted to be members of the positive or negative class depending on which half-space they are located in with respect to the separating hyperplane. The enormous impact of Support Vector Machines was triggered by the observation that the dual form of the Support Vector Machine optimization problem only accesses the data points by means of inner products [60], and that this inner product could be replaced by any other inner product, that is by another kernel function.

Over the following decade, a multitude of applications of Support Vector Machines in bioinformatics emerged, which can be divided into three large branches: SVM applications on DNA/RNA sequences, proteins, and gene expression profiles. These branches differ in the biological objects or data types that they study, but they often make use the of same computational techniques. String kernels, for example, can be applied both to DNA/RNA and protein sequences.

15.5.1 DNA and RNA Sequences

Classification of DNA and RNA sequences via Support Vector Machines is one of the prime applications of SVMs in computational biology.

15.5.1.1 DNA Sequences

Several SVM-based prediction problems on DNA sequences have been studied in the literature, including secondary structure prediction from DNA sequence by an RBF kernel [25], but gene finding is the central prediction task on genomic sequences that SVMs have been applied to over recent years.

Support Vector Machines were successfully applied to various tasks in gene finding, in particular for splice site recognition. The prediction task is here to discriminate between sequences that do contain a true splice site versus sequences with a decoy splice site [73]. The string kernel employed is the weighted degree shift kernel. It builds upon the spectrum kernel, counting matching n-mers in two strings, but the n-mers must occur at similar positions within in the sequence, not at arbitrary positions as in the spectrum kernel. Multiple kernel learning techniques were employed in Sonnenburg et al. [76] to determine the sequence motifs that are predictive of true splice sites (see also Sect. 15.6.2). In Ratsch et al. [56], this technique was further extended to the recognition of alternatively spliced exons. It was applied both to known exons to detect alternatively spliced ones, and to introns in order to check whether they might contain a yet unknown alternatively spliced exon. In Sonnenburg et al. [78], SVMs were employed for promoter recognition in humans. The SVM used a combination of kernels on weak indicators of promoter presence, including strings kernels on specific sequence motifs and properties and a linear kernel on the stacking energy and the twistedness of the DNA. These algorithmic components were assembled into a complete system for gene finding that was used to assay and improve the accuracy of the genome annotation of the nematode *Caenorhabditis elegans* [57], correctly identifying all exons and introns in 87% (coding and untranslated regions) and 95% (coding regions only) of all genes tested in several out-of-sample evaluations. A kernel-based approach was also presented for the identification of regulatory modules in euchromatic sequences [64]. The prediction task is here to decide whether a promoter region is the target of a transcription factor or not. The kernel designed for this task compares the sequence region around the best matches of a set of motifs within the sequence and their relative positions to the transcription start site.

15.5.1.2 RNA Sequences

Support Vector Machines have also been applied in RNA research. A major classification problem that arises in this field is to decide whether an RNA sequence is member of a functional RNA family. For this task, special-purpose kernels on RNA sequences have been defined, so-called stem kernels, which compare the stem structures that appear in the secondary structure of two RNA sequences [58, 59]. The stem kernel examines all possible common base pairs and stem structures of arbitrary lengths, including pseudoknots between two RNA sequences, and calculates the inner product of common stem structure counts. Other typical applications of SVMs in RNA research include distinguishing protein-coding from non-coding RNA [42] and predicting target genes for microRNAs [31,92].

15.5.2 Proteins

A second large area of SVM applications in biology is proteomics, in particular in protein structure, function and interaction prediction.

15.5.2.1 Protein Sequence Comparison

Protein comparison tries to establish the similarity of two proteins in order to find proteins that belong to the same structural or functional class. This comparison can focus on different aspects of the protein: its amino acid sequence, (approximated) physicochemical properties, or its 3D structure.

Comparing and classifying protein sequences is one of the classic tasks in bioinformatics, and one step towards goals such as protein function prediction, protein structure prediction, fold recognition, or remote homology detection. Kernels on sequences in combination with Support Vector Machines contributed to the field of sequence comparison by enabling discriminative classification of sequences. This field in kernel machines in bioinformatics witnessed a lot of work on kernel design, resulting in a number of conceptually different kernels, which we describe in the following.

The *Fisher kernel* combines Support Vector Machines with Hidden Markov Models for protein remote homology detection [27]. The Hidden Markov Model is trained on protein sequences from the positive class and then applied to all proteins in the training and test set to derive a feature vector representation of the protein in terms of a gradient vector. This Fisher-kernel – used within a SVM – outperformed classic sequence alignment techniques such as BLAST [1] in protein homology detection. The Fisher-kernel was later generalised to the class of marginalised kernels on sequences [82]: these kernels apply to all objects that are generated from latent variable models (e.g., HMM). The central idea is to first define a joint kernel for the complete data which includes both visible and hidden variables. The marginalized kernel for visible data is then obtained by taking the expectation with respect to the hidden variables.

Ding and Dubchak [14] derived feature vector representations of the physicochemical properties of proteins from their amino acid sequence and then used these vectors, a kernel on vectors and SVMs to predict SCOP fold membership of proteins [47]. The physicochemical properties for these *composition kernels* were derived by means of amino acid indices [30]: These indices are tables which map each amino acid type to one scalar that approximately describes a physicochemical property of this amino acid, for instance, its polarity, polarizability, van der Waals volume, or hydrophobicity. Cai et al. [13] used a similar approach to classify proteins into structural classes.

Motif kernels, as defined by Logan et al. [43] and Ben-Hur and Brutlag [2], are an alternative way of representing a protein sequence by a vector whose components indicate motif occurrence or absence. Logan et al. [43] use weight matrix motifs from the BLOCKS database [23], which are derived from multiple sequence alignments and occur in highly conserved, and often functionally important, regions of the proteins. These motifs are compared to proteins and the resulting scores are used as feature vector representations of the proteins. Ben-Hur and Brutlag [2] employ motifs from the eBLOCKS database of discrete sequence motifs [80], and show how to efficiently compute the resulting motif kernel using a trie data structure.

Liao and Noble [41] defined a different feature vector representation of protein sequence, resulting in an *empirical kernel* that directly uses existing sequence alignment techniques: For a set of n proteins, they first compute a $n \times n$ matrix of sequence similarity scores (for instance, Smith-Waterman scores [70]) and then represent each protein by its corresponding vector of sequence similarity scores in this matrix.

The most recent class of protein sequence kernels are *string kernels* that count common substrings in two strings (see Sect. 15.4). These kernels either require exact matches [36], allow for a limited number of mismatches [39], or allow for substitutions, gaps or wildcards [38].

Further kernels on sequences have been defined which take local properties of the sequence [44] and local alignments [88] into account for specific prediction tasks, such as subcellular localisation prediction.

15.5.2.2 Protein Structure Comparison

With the ability to determine protein structure more rapidly advancing than our ability to study function, function predictions from protein structure gained more and more attention in computational biology. Dobson and Doig [15] described 1,178 protein structures as vectors by means of simple features such as secondary-structure content, amino acid propensities, surface properties and ligands, to then classify them into enzymes and non-enzymes via Support Vector Machines. Borgwardt et al. [9] modeled proteins from the same dataset as graphs, in which nodes represent secondary structure elements and edges represent neighborship of these elements along the amino acid chain or in 3D space. They then employed a random walk graph kernel on these graph models to perform function prediction and improved over the results achieved by Dobson and Doig [15]. On other benchmark datasets for functional and structural classification, Qiu et al. [52] showed that a kernel that employs similarity scores based on the structural alignment tool MAMMOTH [51] outperforms the previous vector- and graph-based approaches.

15.5.2.3 Protein Interaction Prediction

A third central topic in computational proteomics is the prediction of protein–protein interactions, due to the numerous false-positive and false-negative edges in currently known protein–protein interaction networks. This problem can be cast as a binary classification problem: a pair of proteins is predicted to interact (positive class) or not (negative class). Bock and Gough [5] defined the first Support Vector Machine approach to this problem, in which they represented each pair of proteins as a concatenated feature vector of physicochemical and surface properties of these two proteins. Ben-Hur and Noble [3] further refined this approach by defining a pairwise tensor kernel k_{tensor} on two pairs of proteins (a,b) and (c,d):

$$k_{tensor}((a,b),(c,d)) = k_{single}(a,c)k_{single}(b,d) + k_{single}(b,c)k_{single}(a,d), \quad (15.5)$$

where k_{single} measures the similarity between two proteins based on their sequences, gene ontology annotations, local properties of the network, and homologous interactions in other species. Two pairs of proteins are similar in this kernel, if for each protein in one pair, a protein with similar properties can be found in the other pair.

A setback of the tensor product kernels is the fact that the similarity or dissimilarity of the proteins *within* one pair is not taken into account. This changed when the metric learning pairwise kernel k_{mlpk} was defined [87]:

$$k_{mlpk}((a,b),(c,d)) = [(\phi(a) - \phi(b))'(\phi(c) - \phi(d))]^2, \tag{15.6}$$

which directly compares the relative similarity of the two proteins, $(\phi(a) - \phi(b))$ and $(\phi(c) - \phi(d))$, to each other and improves upon the prediction accuracy of the tensor kernel.

The pairwise tensor kernel and a Gaussian Radial Basis Function (RBF) kernel that considers within-pair-similarity of proteins were used in a recent study to predict co-complex-membership of protein pairs in yeast [53]. The tensor kernel was based on a kernel k_{single}, a weighted sum of kernels including three kernels on protein sequences and three diffusion kernels which measure proximity of the proteins within a physical or genetic interaction network. The Gaussian RBF kernel was computed on features that reflect coexpression, coregulation, colocalisation, similar gene ontology annotation and interologs of the proteins within a pair.

All the kernel methods for protein-interaction prediction via SVMs have in common that they treat the existence of interactions as pairwise independent events, that is, the existence of one interaction does not make the existence of other interactions more or less likely.

15.5.2.4 Other Kernel Applications in Proteomics

Other applications of SVMs in proteomics mainly involve protein function prediction from data sources other than sequence or structure, for which we describe some representative examples here. In one of the early studies in this direction, [86] defines a kernel on trees for function prediction from phylogenetic profiles of proteins. Tsuda and Noble [83] present an approach for predicting the function of unannotated proteins in protein-interaction or metabolomic networks. Their method uses a locally constrained diffusion kernel, which maximises the von Neumann entropy network, to measure similarity between nodes, and a Support Vector Machine for annotating proteins with unknown function.

15.5.3 Gene Expression Profiles

Another popular field of SVM applications are predictions based on microarray gene expression measurements. Existing kernels on vectors, such as the linear, polynomial and Gaussian RBF kernel, can be readily applied here without involved kernel design.

15.5.3.1 Diagnosis and Prognosis

The most common task in this field is to predict the phenotype of a patient based on his or her gene expression levels, primarily for disease diagnosis or for drug response prediction. The first study of this kind was conducted by Mukherjee et al. [46] on the dataset of gene expression levels of two classes of leukemia patients from Golub et al. [18], to tell apart these two subtypes of leukemia using a linear kernel and a SVM. Many similarly interesting studies followed, each of them focusing on a particular task of diagnosis or prognosis. The first kernel for *time series* of microarrays was defined in Borgwardt et al. [10]. Here, gene expression profiles of multiple sclerosis patients were compared to predict their response to treatment by the drug beta-interferon by means of a dynamical systems kernel [90].

15.5.3.2 Function Prediction

SVMs on gene expression levels were also used for gene function prediction. Here, a gene is represented as a vector of its expression levels across different conditions, tissues or patients. The underlying assumption is that two genes are functionally related if they exhibit similar expression levels under different external conditions. The first study is this direction [12] predicted the membership of 6,000 yeast genes to five functional classes from the MIPS Yeast Genome Database [45].

15.6 Kernel Methods Beyond Classification

While Support Vector Machines are clearly the most popular kernel method in bioinformatics, there are also learning problems in bioinformatics which require different algorithmic machinery and statistical tests than classification or regression.

15.6.1 Data Integration for Network Inference

First, several kernel methods for data integration, in particular on networks, were defined.

Kato et al. [29] model protein interaction prediction as a kernel matrix completion problem. Their setting is that they are given a large dataset of proteins with different types of information on these proteins, including gene expression levels, protein localization, and phylogenetic profiles. They represent each of these data types by a 'large' kernel matrix. They are also given the true protein interactions between a small subset of all proteins, which they convert into a 'small' kernel matrix. They then define an algorithm for completing the small kernel matrix by means of the information from the large kernel matrices and to thereby infer the missing, unknown interactions.

Yamanishi et al. [95] also define a supervised approach to protein network inference from multiple types of data including gene expression, localisation information and phylogenetic profiles. They combine ideas from spectral clustering and kernel canonical correlation analysis to derive features that are indicative of protein interaction. This technique is further refined in Yamanishi et al. [96] for enzyme network inference by enforcing chemical constraints to be fulfilled by the resulting network structure.

15.6.2 Feature Selection

Second, feature selection is an important problem in computational biology, as the features that are relevant for an accurate prediction are extremely important to understand the underlying biological process.

A typical example for the relevance of feature selection in bioinformatics is gene selection from microarray data. Support Vector Machine-based approaches to feature selection were defined early on, which recursively eliminate irrelevant features [21] or iteratively downscale the less informative features [94].

Borgwardt et al. [9] and Sonnenburg et al. [76] employed multiple kernel learning for feature selection, to weight different kernels used by a Support Vector Machine. In [9], hyperkernels were used to determine which node attributes in a graph model of protein structure were most important for correct protein function prediction. These nodes represented alpha-helices or beta-sheets in the tertiary structure of the protein, and their attributes were their length in amino acids and Angstroms, and statistics on their hydrophobicity, polarity, polarizability and van der Waals volume. Among all these attributes, hyperkernel learning assigned the largest weight to the amino acid length.

In [76], multiple kernel learning was used to determine those sequence motifs that are most relevant for correct splice site recognition. Each kernel represented one single sequence motif at a specific sequence position, and multiple kernel learning determined the weight for each of these motifs, resulting in a set of position-specific sequence patterns that are associated with true splice sites. This technique was further refined in Sonnenburg et al. [77], now taking the overlap in sequence between different substrings into account and allowing to assess the importance of (consensus) sequence motifs for correct prediction, even if they do not occur in the given collection of sequences.

Song et al. [71] define a kernel-based approach to gene selection from microarray data. They show that many of the vast number of feature selection algorithms from the microarray literature are indeed instances of this framework, which are obtained by a different choice of kernel and/or a particular type of normalisation. New gene selection algorithms can easily be derived from this framework, even for regression and multi-class settings, and existing techniques can be objectively compared to each other, by replacing one kernel by another, while keeping other properties fixed, such as the normalisation technique employed.

15.6.3 Statistical Tests

Third, a recent development in machine learning are kernel-based statistical tests [19,20], which led to a first application in bioinformatics: Borgwardt [7] define a kernel-based statistical test to check cross-platform comparability of microarray data. This two-sample test, whose goal it is to establish whether two samples were drawn from the same distribution or not, computes the distance between the means of the two samples in a universal reproducing kernel Hilbert Space [79] as its test statistic. The larger this distance, the smaller the probability that the two samples originate from the same distribution. In experiments on microarray cross-platform comparability, the test manages to clearly distinguish between samples of microarray measurements that were generated on the same platform and those from different platforms.

15.6.4 Kernel Methods for Structured Output

Fourth, another recent development in kernel machine learning are kernel methods for structured output domains. The classic Support Vector Machine was designed for binary classification problems, and data objects that were drawn i.i.d (independently and identically distributed) from an underlying distribution. However, it is obvious that many prediction problems in biology are multi-class problems, and that predictions on different objects can depend highly on each other.

For instance, if one wants to annotate a DNA sequence in gene finding, the predicted label of a nucleotide (e.g., exonic or intronic) is highly dependent on those of the neighbouring bases. This is often referred to as the *label sequence learning problem*: Given a sequence of n letters, one wants to predict a sequence of n class labels. Hidden Markov Models are the classic tool for this problem in computational biology [16]. Conditional random fields were developed as a discriminative alternative to the generative model that Hidden Markov Models are based upon Lafferty et al. [33]. Kernel-based discriminative approaches to this problem have recently been defined in machine learning as well, and employed successfully for sequence alignment [6, 65], gene finding and genome annotation [57, 66], and tiling array analysis [97, 98]. A general approach to Support Vector Machine classification in multiclass and structured output domains was proposed by Tsochantaridis et al. [81], and promises to trigger further research in this direction in computational biology.

15.6.5 Outlook

In our opinion, the success story of kernel methods in bioinformatics will continue over the next decade. The strength of kernels in dealing with structured objects will lead to more applications of kernels in biological network analysis. Their ability to elegantly handle high-dimensional data and to integrate various data sources will make them one attractive tool for tasks such as genome-wide association studies. Furthermore, the ability to encode prior knowledge in the kernel function will foster the use of kernel methods in various specialised prediction tasks in computational biology.

References

1. Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, *215*(3), 403–410.
2. Ben-Hur, A., & Brutlag, D. (2003). Remote homology detection: A motif based approach. *Bioinformatics*, *19* (Suppl. 1), i26–i33. URL http://www.ncbi.nlm.nih.gov/pubmed/12855434. PMID: 12855434
3. Ben-Hur, A., & Noble, W. S. (2005). Kernel methods for predicting protein-protein interactions. *Bioinformatics (Oxford, England)*, *21* (Suppl. 1), i38–i46. DOI 10.1093/bioinformatics/bti1016. URL http://www.ncbi.nlm.nih.gov/pubmed/15961482. PMID: 15961482
4. Ben-Hur, A., Ong, C. S., Sonnenburg, S., Schölkopf, B., & Rätsch, G. (2008). Support vector machines and kernels for computational biology. *PLoS Computational Biology*, *4*(10), e1000,173. DOI 10.1371/journal.pcbi.1000173. URL http://www.ncbi.nlm.nih.gov/pubmed/18974822. PMID: 18974822
5. Bock, J. R., & Gough, D. A. (2001). Predicting protein–protein interactions from primary structure. *Bioinformatics (Oxford, England)*, *17*(5), 455–460. URL http://www.ncbi.nlm.nih.gov/pubmed/11331240. PMID: 11331240
6. Bona, F. D., Ossowski, S., Schneeberger, K., & Rätsch, G. (2008). Optimal spliced alignments of short sequence reads. *Bioinformatics (Oxford, England)*, *24*(16), i174–i180. DOI 10.1093/bioinformatics/btn300. URL http://www.ncbi.nlm.nih.gov/pubmed/18689821. PMID: 18689821
7. Borgwardt, K. M., Gretton, A., Rasch, M. J., Kriegel, H. P., Schölkopf, B., & Smola, A. J. (2006). Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics (ISMB)*, *22*(14), e49–e57.
8. Borgwardt, K. M., & Kriegel, H. P. (2005). Shortest-path kernels on graphs. In *ICDM* (pp. 74–81). IEEE Computer Society.
9. Borgwardt, K. M., Ong, C. S., Schönauer, S., Vishwanathan, S. V. N., Smola, A. J., & Kriegel, H. P. (2005). Protein function prediction via graph kernels. *Bioinformatics*, *21*(Suppl 1), i47–i56.
10. Borgwardt, K. M., Vishwanathan, S. V. N., & Kriegel, H. P. (2006). Class prediction from time series gene expression profiles using dynamical systems kernels. In R. B. Altman, T. Murray, T. E. Klein, A. K. Dunker, & L. Hunter (Eds.), *Pacific symposium on biocomputing* (pp. 547–558). World Scientific.
11. Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In D. Haussler (Ed.), *Proceedings of the annual conference on computational learning theory* (pp. 144–152). Pittsburgh, PA: ACM.

12. Brown, M. P. S., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C., Furey, T. S., et al. (2000). Knowledge-based analysis of microarray gene expression data using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America*, *97*(1), 262–267.
13. Cai, Y. D., Liu, X. J., Xu, X. B., & Chou, K. C. (2002). Prediction of protein structural classes by support vector machines. *Computational Chemistry*, *26*(3), 293–296.
14. Ding, C. H., & Dubchak, I. (2001). Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics*, *17*(4), 349–358.
15. Dobson, P. D., & Doig, A. J. (2003). Distinguishing enzyme structures from non-enzymes without alignments. *Journal of Molecular Biology*, *330*(4), 771–783.
16. Durbin, R., Eddy, S., Krogh, A., & Mitchison, G. (1998). *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge, UK: Cambridge University Press.
17. Gärtner, T., Flach, P. A., & Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. In B. Schölkopf & M. K. Warmuth (Eds.), *COLT*, *Lecture Notes in Computer Science* (Vol. 2777, pp. 129–143). Springer.
18. Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., et al. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, *286*(5439), 531–537.
19. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., & Smola, A. (2007). A kernel method for the two-sample-problem. In *Advances in neural information processing systems* (Vol. 19, pp. 513–520). Cambridge, MA: MIT.
20. Gretton, A., Fukumizu, K., Teo, C. H., Song, L., Schölkopf, B., & Smola, A. J. (2007). A kernel statistical test of independence. In J. C. Platt, D. Koller, Y. Singer, & S. T. Roweis (Eds.), *NIPS*. MIT Press.
21. Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*, 389–422.
22. Haussler, D. (1999). *Convolutional kernels on discrete structures*. Tech. Rep., UCSC-CRL-99-10. UC Santa Cruz: Computer Science Department.
23. Henikoff, S., Henikoff, J. G. (1991). Automated assembly of protein blocks for database searching. *Nucleic Acids Research*, *19*, 6565–6572.
24. Horváth, T., Gärtner, T., & Wrobel, S. (2004). Cyclic pattern kernels for predictive graph mining. In W. Kim, R. Kohavi, J. Gehrke, & W. DuMouchel (Eds.), *KDD* (pp. 158–167). ACM.
25. Hua, S., & Sun, Z. (2001). A novel method of protein secondary structure prediction with high segment overlap measure: Support vector machine approach. *Journal of Molecular Biology*, *308*(2), 397–407. DOI 10.1006/jmbi.2001.4580. URL http://www.ncbi.nlm.nih.gov/pubmed/11327775. PMID: 11327775
26. Imrich, W., & Klavzar, S. (2000). Product graphs: Structure and recognition. In *Wiley Interscience Series in Discrete Mathematics*. New York: Wiley VCH.
27. Jaakkola, T., Diekhans, M., & Haussler, D. (1999). Using the fisher kernel method to detect remote protein homologies. In T. Lengauer, R. Schneider, P. Bork, D. L. Brutlag, J. I. Glasgow, H. W. Mewes, et al. (Eds.), *ISMB* (pp. 149–158). AAAI.
28. Kashima, H., Tsuda, K., & Inokuchi, A. (2003). Marginalized kernels between labeled graphs. In *Proceedings of the* 20*th International Conference on Machine Learning (ICML)*. Washington, DC: United States.
29. Kato, T., Tsuda, K., & Asai, K. (2005). Selective integration of multiple biological data for supervised network inference. *Bioinformatics (Oxford, England)*, *21*(10), 2488–2495. DOI 10.1093/bioinformatics/bti339. URL http://www.ncbi.nlm.nih.gov/pubmed/15728114. PMID: 15728114
30. Kawashima, S., Ogata, H., & Kanehisa, M. (1999). Aaindex: Amino acid index database. *Nucleic Acids Research*, *27*(1), 368–369.
31. Kim, S., Nam, J., Rhee, J., Lee, W., & Zhang, B. (2006). miTarget: microRNA target gene prediction using a support vector machine. *BMC Bioinformatics*, *7*, 411. DOI 10.1186/1471-2105-7-411. URL http://www.ncbi.nlm.nih.gov/pubmed/16978421. PMID: 16978421

32. Kuksa, P. P., Huang, P. H., & Pavlovic, V. (2008). Scalable algorithms for string kernels with inexact matching. In D. Koller, D. Schuurmans, Y. Bengio, & L. Bottou (Eds.), *NIPS* (pp. 881–888). MIT.
33. Lafferty, J. D., McCallum, A., & Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In C. E. Brodley & A. P. Danyluk (Eds.), *ICML* (pp. 282–289). Morgan Kaufmann.
34. Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L. E., & Jordan, M. I. (2004). Learning the kernel matrix with semi-definite programming. *Journal of Machine Learning Research*, *5*, 27–72.
35. Lanckriet, G. R. G., Bie, T. D., Cristianini, N., Jordan, M. I., & Noble, W. S. (2004). A statistical framework for genomic data fusion. *Bioinformatics*, *20*(16), 2626–2635. DOI 10.1093/bioinformatics/bth294. URL http://www.ncbi.nlm.nih.gov/pubmed/15130933. PMID: 15130933
36. Leslie, C., Eskin, E., & Noble, W. S. (2002). The spectrum kernel: A string kernel for SVM protein classification. In *Proceedings of the pacific symposium on biocomputing* (pp. 564–575).
37. Leslie, C., Eskin, E., Weston, J., & Noble, W. S. (2002). Mismatch string kernels for SVM protein classification. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in neural information processing systems* (Vol. 15). Cambridge, MA: MIT.
38. Leslie, C. S., & Kuang, R. (2003). Fast kernels for inexact string matching. In B. Schölkopf & M. K. Warmuth (Eds.), *COLT*, *Lecture Notes in Computer Science* (Vol. 2777, pp. 114–128). Springer.
39. Leslie, C. S., Eskin, E., Cohen, A., Weston, J., & Noble, W. S. (2004). Mismatch string kernels for discriminative protein classification. *Bioinformatics (Oxford, England)*, *20*(4), 467–476. DOI 10.1093/bioinformatics/btg431. URL http://www.ncbi.nlm.nih.gov/pubmed/14990442. PMID: 14990442
40. Lewis, D. P., Jebara, T., & Noble, W. S. (2006). Support vector machine learning from heterogeneous data: An empirical analysis using protein sequence and structure. *Bioinformatics (Oxford, England)*, *22*(22), 2753–2760. DOI 10.1093/bioinformatics/btl475. URL http://www.ncbi.nlm.nih.gov/pubmed/16966363. PMID: 16966363
41. Liao, L., & Noble, W. S. (2002). Combining pairwise sequence similarity and support vector machines for remote protein homology detection. In *RECOMB* (pp. 225–232).
42. Liu, J., Gough, J., & Rost, B. (2006). Distinguishing Protein-Coding from Non-Coding RNAs through support vector machines. *PLoS Genetics*, *2*(4), 529–536.
43. Logan, B., Moreno, P., Suzek, B., Weng, Z., & Kasif, S. (2001). *A study of remote homology detection*. Tech. Rep., Cambridge Research Laboratory.
44. Matsuda, S., Vert, J., Saigo, H., Ueda, N., Toh, H., & Akutsu, T. (2005). A novel representation of protein sequences for prediction of subcellular location using support vector machines. *Protein Science: A Publication of the Protein Society*, *14*(11), 2804–2813. DOI 10.1110/ps.051597405. URL http://www.ncbi.nlm.nih.gov/pubmed/16251364. PMID: 16251364
45. Mewes, H. W., Frishman, D., Gruber, C., Geier, B., Haase, D., Kaps, A., et al. (2000). MIPS: A database for genomes and protein sequences. *Nucleic Acids Research*, *28*(1), 37–40. URL http://www.ncbi.nlm.nih.gov/pubmed/10592176. PMID: 10592176
46. Mukherjee, S., Tamayo, P., Slonim, D., Verri, A., Golub, T., Mesirov, J.P., et al. (2000). *Support vector machine classification of microarray data*. Tech. Rep., Artificial Intelligence Laboratory, Massachusetts Institute of Technology.
47. Murzin, A. G., Brenner, S. E., Hubbard, T., & Chothia, C. (1995). SCOP: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, *247*(4), 536–40. DOI 10.1006/jmbi.1995.0159. URL http://www.ncbi.nlm.nih.gov/pubmed/7723011. PMID: 7723011
48. Noble, W. (2004). Support vector machine applications in computational biology. In B. Schölkopf, K. Tsuda, & J. P. Vert (Eds.), *Kernel methods in computational biology*. Cambridge, MA: MIT.
49. Noble, W. S. (2006). What is a support vector machine? *Nature Biotechnology*, *24*(12), 1565–1567. DOI 10.1038/nbt1206-1565. URL http://dx.doi.org/10.1038/nbt1206-1565

50. Ong, C. S., & Smola, A. J. (2003). Machine learning with hyperkernels. In T. Fawcett & N. Mishra (Eds.), *ICML* (pp. 568–575). AAAI.
51. Ortiz, A. R., Strauss, C. E. M., & Olmea, O. (2002). MAMMOTH (matching molecular models obtained from theory): An automated method for model comparison. *Protein Science: A Publication of the Protein Society*, *11*(11), 2606–2621. DOI 10.1110/ps.0215902. URL http://www.ncbi.nlm.nih.gov/pubmed/12381844. PMID: 12381844
52. Qiu, J., Hue, M., Ben-Hur, A., Vert, J., & Noble, W. S. (2007). A structural alignment kernel for protein structures. *Bioinformatics (Oxford, England)*, *23*(9), 1090–1098. DOI 10.1093/bioinformatics/btl642. URL http://www.ncbi.nlm.nih.gov/pubmed/17234638. PMID: 17234638
53. Qiu, J., & Noble, W. S. (2008). Predicting co-complexed protein pairs from heterogeneous data. *PLoS Computational Biology*, *4*(4), e1000,054. DOI 10.1371/journal.pcbi.1000054. URL http://www.ncbi.nlm.nih.gov/pubmed/18421371. PMID: 18421371
54. Rakotomamonjy, A., Bach, F., Canu, S., & Grandvalet, Y. (2007). More efficiency in multiple kernel learning. In Z. Ghahramani (Ed.), *ICML*, *ACM International Conference Proceeding Series* (Vol. 227, pp. 775–782). ACM.
55. Ramon, J., & Gärtner, T. (2003). *Expressivity versus efficiency of graph kernels*. Tech. Rep., First International Workshop on Mining Graphs, Trees and Sequences (held with ECML/PKDD'03).
56. Rätsch, G., Sönnenburg, S., & Schölkopf, B. (2005). RASE: Recognition of alternatively spliced exons in *c. elegans*. *Bioinformatics*, *21* (Suppl. 1), i369–i377.
57. Rätsch, G., Sonnenburg, S., Srinivasan, J., Witte, H., Müller, K., Sommer, R., et al. (2007). Improving the *Caenorhabditis elegans* genome annotation using machine learning. *PLoS Computational Biology*, *3*(2), e20. PMID: 17319737
58. Sakakibara, Y., Popendorf, K., Ogawa, N., Asai, K., & Sato, K. (2007). Stem kernels for RNA sequence analyses. *Journal of Bioinformatics and Computational Biology*, *5*(5), 1103–1122. URL http://www.ncbi.nlm.nih.gov/pubmed/17933013. PMID: 17933013
59. Sato, K., Mituyama, T., Asai, K., & Sakakibara, Y. (2008). Directed acyclic graph kernels for structural RNA analysis. *BMC Bioinformatics*, *9*, 318. DOI 10.1186/1471-2105-9-318. URL http://www.ncbi.nlm.nih.gov/pubmed/18647390. PMID: 18647390
60. Schölkopf, B. (1997). *Support vector learning*. München: R. Oldenbourg Verlag. PhD thesis, TU Berlin. Download: http://www.kernel-machines.org
61. Schölkopf, B., & Smola, A. J. (2002). *Learning with Kernels*. Cambridge, MA: MIT.
62. Schölkopf, B., Smola, A. J., & Müller, K. R. (1997). Kernel principal component analysis. In W. Gerstner, A. Germond, M. Hasler, & J. D. Nicoud (Eds.), *Artificial neural networks ICANN'97* (Vol. 1327, pp. 583–588). Berlin: Springer Lecture Notes in Computer Science.
63. Schölkopf, B., Tsuda, K., & Vert, J. P. (2004). *Kernel Methods in Computational Biology*. Cambridge, MA: MIT.
64. Schultheiss, S. J., Busch, W., Lohmann, J. U., Kohlbacher, O., & Rätsch, G. (2009). KIRMES: kernel-based identification of regulatory modules in euchromatic sequences. *Bioinformatics (Oxford, England)*, DOI 10.1093/bioinformatics/btp278. URL http://www.ncbi.nlm.nih.gov/pubmed/19389732. PMID: 19389732
65. Schulze, U., Hepp, B., Ong, C. S., & Rätsch, G. (2007). PALMA: mRNA to genome alignments using large margin algorithms. *Bioinformatics (Oxford, England)*, *23*(15), 1892–1900. DOI 10.1093/bioinformatics/btm275. URL http://www.ncbi.nlm.nih.gov/pubmed/17537755. PMID: 17537755
66. Schweikert, G., Zien, A., Zeller, G., Behr, J., Dieterich, C., Ong, C. S., et al. (2009). mGene: Accurate SVM-based gene finding with an application to nematode genomes. *Genome Research*, *19*(11), 2133–2143. DOI 10.1101/gr.090597.108. URL http://www.ncbi.nlm.nih.gov/pubmed/19564452. PMID: 19564452
67. Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge, UK: Cambridge University Press.
68. Shervashidze, N., & Borgwardt, K. M. (2009). Fast subtree kernels on graphs. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, & A. Culotta (Eds.), *NIPS* (pp. 1660–1668). Cambridge, MA: MIT.

69. Shervashidze, N., Vishwanathan, S., Petri, T., Mehlhorn, K., & Borgwardt, K. M. (2009). Efficient graphlet kernels for large graph comparison. In D. van Dyk & M. Welling (Eds.), *Proceedings of the twelfth international conference on artificial intelligence and statistics*. Clearwater Beach, Florida.
70. Smith, T. F., & Waterman, M. S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, *147*(1), 195–197. URL http://www.ncbi.nlm.nih.gov/pubmed/7265238. PMID: 7265238
71. Song, L., Bedo, J., Borgwardt, K., Gretton, A., & Smola, A. (2007). Gene selection via the BAHSIC family of algorithms. *Bioinformatics*, *23*(13), i490–i498.
72. Song, L., Smola, A., Gretton, A., Borgwardt, K., & Bedo, J. (2007). Supervised feature selection via dependence estimation. In: Ghahramani, Z. (ed.): *ACM International Conference Proceeding Series*, vol. 227. ACM.
73. Sonnenburg, S., Rätsch, G., Jagota, A. K., & Müller, K. R. (2002). New methods for splice site recognition. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)* (pp. 329–336).
74. Sonnenburg, S., Rätsch, G., & Rieck, K. (2007). Large-scale learning with string kernels. In L. Bottou, O. Chapelle, D. DeCoste, & J. Weston (Eds.), *Large-Scale kernel machines* (pp. 73—104). Cambridge, MA: MIT.
75. Sonnenburg, S., Rätsch, G., & Schäfer, C. (2005). A general and efficient multiple kernel learning algorithm. In *NIPS*.
76. Sonnenburg, S., Rätsch, G., & Schäfer, C. (2005). Learning interpretable SVMs for biological sequence classification. In *RECOMB 2005, LNBI 3500* (pp. 389–407). Berlin, Heidelberg: Springer-Verlag.
77. Sonnenburg, S., Zien, A., Philips, P., & Rätsch, G. (2008). POIMs: positional oligomer importance matrices — understanding support vector machine based signal detectors. *Bioinformatics*, *24*(13), i6–i14. URL http://bioinformatics.oxfordjournals.org/cgi/content/full/24/13/i6
78. Sonnenburg, S., Zien, A., & Rätsch, G. (2006). ARTS: Accurate recognition of transcription starts in human. *Bioinformatics (Oxford, England) 22*(14), e472–480. DOI 10.1093/DOI bioinformatics/btl250. URL http://www.ncbi.nlm.nih.gov/pubmed/16873509. PMID: 16873509
79. Steinwart, I. (2002). Support vector machines are universally consistent. *Journal of Complexity*, *18*, 768–791.
80. Su, Q. J., Lu, L., Saxonov, S., & Brutlag, D. L. (2005). eBLOCKs: Enumerating conserved protein blocks to achieve maximal sensitivity and specificity. *Nucleic Acids Research*, *33*(Database issue), D178–D182. DOI 10.1093/nar/gki060. URL http://www.ncbi.nlm.nih.gov/pubmed/15608172. PMID: 15608172
81. Tsochantaridis, I., Joachims, T., Hofmann, T., & Altun, Y. (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, *6*, 1453–1484.
82. Tsuda, K., Kin, T., & Asai, K. (2002). Marginalized kernels for biological sequences. *Bioinformatics (Oxford, England)*, *18* (Suppl. 1), S268–S275. URL http://www.ncbi.nlm.nih.gov/pubmed/12169556. PMID: 12169556
83. Tsuda, K., Noble, W. S. (2004). Learning kernels from biological networks by maximizing entropy. *Bioinformatics (Oxford, England)*, *20* (Suppl. 1), i326–i333. DOI 10.1093/bioinformatics/bth906. URL http://www.ncbi.nlm.nih.gov/pubmed/15262816. PMID: 15262816
84. Tsuda, K., Shin, H., & Schölkopf, B. (2005). Fast protein classification with multiple networks. *Bioinformatics*, *21* (Suppl. 2), ii59–ii65.
85. Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley.
86. Vert, J. (2002). A tree kernel to analyse phylogenetic profiles. *Bioinformatics*, *18*, S276–S284.
87. Vert, J., Qiu, J., & Noble, W. S. (2007). A new pairwise kernel for biological network inference with support vector machines. *BMC Bioinformatics*, *8* (Suppl. 10), S8. DOI 10.1186/1471-2105-8-S10-S8. URL http://www.ncbi.nlm.nih.gov/pubmed/18269702. PMID: 18269702

88. Vert, J. P., Saigo, H., & Akutsu, T. (2004). Local alignment kernels for biological sequences. In B. Schölkopf, K. Tsuda, & J. P. Vert (Eds.), *Kernel methods in computational biology* (pp. 261–274). Cambridge, MA: MIT.
89. Vishwanathan, S., & Smola, A. (2003). Fast kernels for string and tree matching. In K. Tsuda, B. Schölkopf, & J. Vert (Eds.), *Kernels and bioinformatics*. Cambridge, MA: MIT. Forthcoming
90. Vishwanathan, S. V., Smola, A. J., & Vidal, R. (2007). Binet-Cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes. *International Journal of Computer Vision*, *73*(1), 95–119. URL http://portal.acm.org/citation.cfm?id=1227529
91. Vishwanathan, S. V. N., Borgwardt, K., & Schraudolph, N. N. (2007). Fast computation of graph kernels. In B. Schölkopf, J. Platt, & T. Hofmann (Eds.), *Advances in neural information processing systems* (Vol. 19). Cambridge MA: MIT.
92. Wang, X., & Naqa, I. M. E. (2008). Prediction of both conserved and nonconserved microRNA targets in animals. *Bioinformatics (Oxford, England)*, *24*(3), 325–332. DOI 10.1093/bioinformatics/btm595. URL http://www.ncbi.nlm.nih.gov/pubmed/18048393. PMID: 18048393
93. Weinberger, K. Q., Sha, F., & Saul, L. K. (2004). Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the* 21*st international conference on machine learning*. Banff, Canada.
94. Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2000). Feature selection for svms. In T. K. Leen, T. G. Dietterich, V. Tresp (Eds.), *NIPS* (pp. 668–674). MIT.
95. Yamanishi, Y., Vert, J., & Kanehisa, M. (2004). Protein network inference from multiple genomic data: A supervised approach. *Bioinformatics (Oxford, England)*, *20* (Suppl. 1), i363–i370. DOI 10.1093/bioinformatics/bth910. URL http://www.ncbi.nlm.nih.gov/pubmed/15262821. PMID: 15262821
96. Yamanishi, Y., Vert, J., & Kanehisa, M. (2005). Supervised enzyme network inference from the integration of genomic data and chemical information. *Bioinformatics (Oxford, England)*, *21* (Suppl 1), i468–i477. DOI 10.1093/bioinformatics/bti1012. URL http://www.ncbi.nlm.nih.gov/pubmed/15961492. PMID: 15961492
97. Zeller, G., Clark, R. M., Schneeberger, K., Bohlen, A., Weigel, D., & Rätsch, G. (2008). Detecting polymorphic regions in *Arabidopsis thaliana* with resequencing microarrays. *Genome Research*, *18*(6), 918–929.
98. Zeller, G., Henz, S. R., Laubinger, S., Weigel, D., & Rätsch, G. (2008). Transcript normalization and segmentation of tiling array data. In R. B. Altman, A. K. Dunker, L. Hunter, T. Murray, & T.E. Klein (Eds.), *Pacific symposium on biocomputing* (pp. 527–538). World Scientific.

Chapter 16
Graph Classification Methods in Chemoinformatics

Koji Tsuda

Abstract Graphs are general and powerful data structures that can be used to represent diverse kinds of molecular objects such as chemical compounds, proteins, and RNAs. In recent years, computational analysis of tens of thousands of labeled graphs has become possible by advanced graph mining methods. For example, frequent pattern mining methods such as gSpan can enumerate all frequent subgraphs in a graph database efficiently. This chapter reviews basics of graph mining methodology and its application to chemoinformatics and bioinformatics. Graph classification and regression techniques based on subgraph patterns are also reviewed extensively.

16.1 Introduction

Much of the real world data is represented not as vectors, but as graphs including sequences and trees, for example, biological sequences, semi-structured texts such as HTML and XML, chemical compounds, RNA secondary structures, API call graphs and so forth. Recently we have seen a surge of interest in graph data processing. The topic itself is not new. Since 1970s, there has been continuous effort in developing methods for processing such graph data. For instance, graph alignment is a classic example [31]. However, in the beginning of 2000s, the development of graph kernels [16] and graph mining [38] ignited the interests in many fields of computer science. Among them, chemoinformatics is the most prominent field with largest repository of data. For example, NCBI's PubChem has millions of chemical compounds that are naturally represented as molecular graphs. Also, many different kinds of chemical activity data are available, which provides a huge testbed for graph classification methods. In addition, protein 3D structures [4] and RNA secondary structures [12] can naturally be represented as labeled graphs.

K. Tsuda
AIST Computational Biology Research Center Tokyo, Japan
e-mail: koji.tsuda@aist.go.jp

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_16,
© Springer-Verlag Berlin Heidelberg 2011

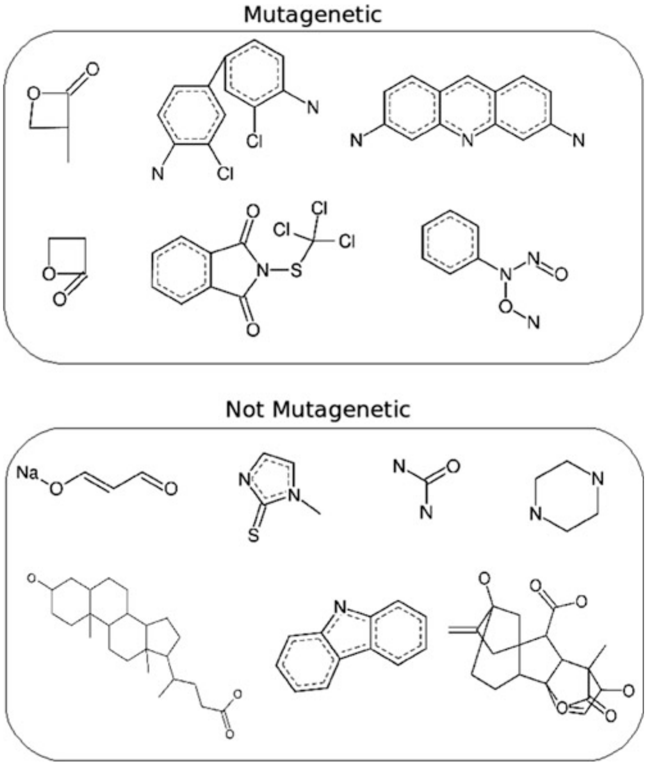


Fig. 16.1 Examples of chemical compounds with and without mutagenicity

Figure 16.1 illustrates an example of chemical compound data. Here these graphs are classified according to their mutagenicity, i.e., the ability of causing mutations in human DNA. There would be many learning tasks from such data, but the following two are among the most important ones.

Frequent Pattern Mining Identify frequently appearing subgraphs (i.e., patterns) in the graphs with mutagenicity.

Graph Classification Construct a prediction rule that classifies yet unseen compounds.

The two tasks are related to each other, namely one can use frequent patterns to construct prediction rules [30]. In chemoinformatics, graph classification is often called the Structure-Activity Relationship (SAR) problem [10].

Frequent pattern mining techniques are main tools in general data mining, not only for graphs [13]. The simplest one is itemset mining [1], where frequent subsets are enumerated from a series of sets. Since the proposal of itemset mining, researchers have been trying to apply the frequent pattern mining to more structured data such as sequences [26] and trees [2]. At the final stage of development, frequent subgraph enumeration algorithms such as AGM [15], Gaston [24] and gSpan [38] were proposed to deal with the most general structure, labeled graphs

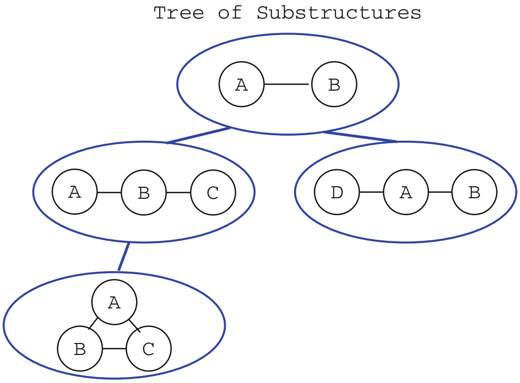


Fig. 16.2 Schematic figure of the tree-shaped search space of graph patterns (i.e., the DFS code tree). To find the optimal pattern efficiently, the tree is systematically expanded by rightmost extensions

with loops. They can enumerate all the subgraph patterns that appear more than m times in a graph database. The threshold m is called *minimum support*. Frequent subgraph patterns are found by branch-and-bound search in a tree shaped search space (Fig. 16.2). The computational time crucially depends on the minimum support parameter. For chemical compound datasets, it is easy to mine tens of thousands of graphs on a usual PC, if the minimum support is reasonably high (e.g., 10% of the number of graphs).

For graph classification, one can just pipeline frequent pattern mining and an existing classification algorithm such as support vector machines [32]. However, to achieve the best accuracy, the minimum support has to be determined to a small value (e.g., smaller than 1%) [14, 17, 36]. In such setting, the graph mining becomes prohibitively inefficient, and creates millions of patterns, which make subsequent processing difficult. Graph boosting [30] progressively constructs the prediction rule in an iterative fashion, and in each iteration only a few informative subgraphs are discovered. In comparison to the naive method using frequent mining and support vector machines, the graph mining routine has to be called multiple times. However, thanks to an additional search tree pruning condition, one call finishes quickly, and the overall time is shorter than the naive method.

Notice that the graph classification is possible without resorting to pattern mining. A prominent approach is the combination of graph kernels and support vector machines [16]. However, this technique is covered by another chapter.

The rest of this chapter is as follows: In Sect. 16.2, we describe basics of pattern mining methods. In Sect. 16.3, the main topic is graph classification and their applications. We conclude the chapter in Sect. 16.4.

16.2 Frequent Pattern Mining

In this section, we aim to provide an intuitive understanding of graph mining methods. For this purpose, it is probably best to trace the history of pattern mining methods. Itemset mining has been extended to more structured data, such as transaction sequences [26], trees [2] and labeled graphs [38]. We first explain itemset mining and then modifications required for mining graphs.

16.2.1 Itemset Mining

Itemset mining is mainly used in business applications such as market basket analysis: In a supermarket, one customer buys several items at the same time, e.g., carrot, milk, potato etc. Given these *transactions*, one would like to know the set of items appearing together frequently. If it turns out that (milk, beer) is a frequent itemset, they should be placed next to each other to improve sales. More concretely, definition of itemsets and their properties are summarized as follows:

1. Given a set S of items, any nonempty subset of S is called an itemset.
2. Given an itemset I and a set T of transactions, the support of I, denoted as $\mathit{support}(I)$, is the number of transactions that contain all the items in I.
3. Given a positive integer α, I is a frequent itemset, if $\mathit{support}(I) \geq \alpha$. We refer to α as the minimum support parameter.

The computational time of itemset mining depends on the number of solutions (i.e., the number of frequent itemsets), which is unknown in advance. If α is too low, it takes a long time to finish. So a practical advice is to set α to a large value initially (e.g., 50% of the number of transactions), and decrease it gradually until you have a moderate number of solutions (e.g., 10,000).

There are several families of algorithms of itemset mining, but the Apriori algorithm is the earliest and the simplest of all. Given four items, all itemsets form a lattice depicted in Fig. 16.3. If a transaction t includes itemset I, then t includes any subset of I. This property, called anti-monotonicity, plays a very important role. It means that, frequent itemsets whose support is above a threshold appears as a connected region in the lattice (i.e., the highlighted region in the figure). Therefore, we can enumerate frequent itemsets by starting from the empty set and traversing the tree by adding items. Whenever an infrequent itemset is found, we do not need to traverse itemsets in its downstream (tree pruning). One can choose either breadth-first search or depth-first search in traversing the lattice. In the former case, the method is called Apriori algorithm [1], otherwise the backtrack algorithm [39]. In Algorithm 1, we present a summary of the Apriori algorithm.

Algorithm 1 The Apriori algorithm for itemset mining

```
$D_1$ = all frequent itemsets of size 1, $k = 1$
while $D_k$ is not empty do
    Take the union of two itemsets in $D_k$. If their size is $k + 1$, add to $D_{k+1}$
    Remove all infrequent itemsets from $D_{k+1}$
    $k = k + 1$
end while
```

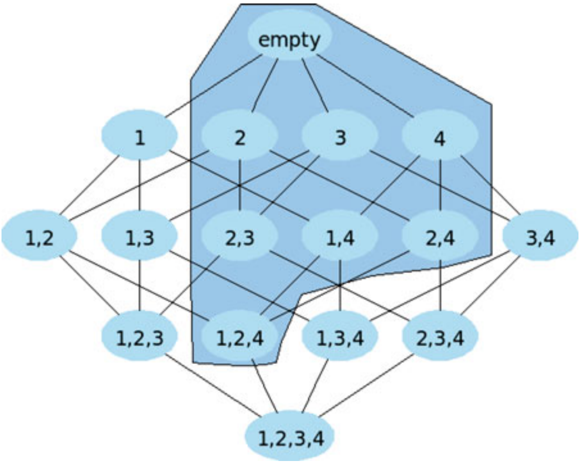


Fig. 16.3 Search tree of Apriori algorithm. The *highlighted* region indicates frequent patterns

Evaluation of computational cost of enumeration algorithms like itemset mining is tricky. Usually, the computational time is measured as a function of input size like $O(n)$. In our case, the input size corresponds to the number of transactions. However, the efficiency of itemset mining crucially depends on the number of solutions. Namely, if the number of frequent itemsets is small, the algorithm finishes quickly, but if there are many frequent itemsets, it takes long time. In worst case evaluation, itemset mining is NP-hard with respect to the number of transactions. However, the worst case means that all transactions have the same set of items. When there are large overlaps in transactions, the dataset is called dense. For dense datasets, itemset mining can be extremely slow. However, in market basket analysis, it is not usually the case, because each customer buys only a small fraction of items. For such sparse data, itemset mining can scale to millions of transactions. For biological applications, it is important to evaluate the density of transactions in advance. If it is too dense, itemset mining might not be a viable choice.

16.2.2 Graph Mining

Like itemset mining, graph mining requires a canonical search space in which a whole set of patterns are traversed without duplication. In gSpan, the DFS (depth first search) code tree is adopted for this purpose. The basic idea of the DFS code tree is to organize patterns as a tree, where a child node has a supergraph of the parent's pattern (Fig. 16.2). In the tree, a pattern is represented as a text string called the DFS code, which is made by traversing the graph by depth first search. Each node is indexed from 0 to $n-1$ according to the discovery time in the DFS. All the edges traversed in the DFS are called forward edges and the rest is called backward edges.

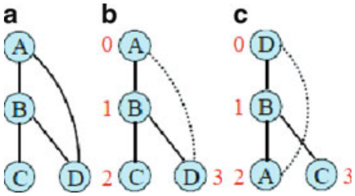


Fig. 16.4 Depth first search and DFS code of graph. (**a**) A graph example. (**b**), (**c**) Two different depth-first-searches of the same graph. *Red* numbers represent the DFS indices. *Bold* edges and *dashed* edges represent the forward edges and the backward edges respectively

One important fact is that, according to the starting node, there are several DFS codes for the same graph (Fig. 16.4). The canonical representation is determined as the minimum code according to the lexicographical order.

The patterns are enumerated by generating the tree from the root to leaves using a recursive algorithm. Node generation is systematically done by rightmost extensions. Still, it is often the case that the same DFS code is generated through different paths. To avoid the duplication, whenever a new node is made, the associated DFS code has to be minimum. It is proven in [38] that, by assuring the minimality of the DFS code in each extension step, the whole set of patterns can be enumerated without duplication. As in the itemset mining, we adopt tree pruning according to the support. If the support of a pattern is found to be smaller than the minimum support threshold, the search tree extension is stopped immediately.

All embeddings of a pattern in the graphs have to be maintained in each node to calculate its support. If a pattern matches a graph in different ways, all such embeddings are stored. When a new pattern is created by adding an edge, it is not necessary to perform full isomorphism checks with respect to all graphs in the database. A new list of embeddings is made by extending the embeddings of the parent. Technically, it is necessary to devise a data structure such that the embeddings are stored incrementally, because it takes a prohibitive amount of memory to keep all embeddings independently in each node.

The most time consuming part of gSpan is the minimality check of the DFS code. It is as expensive as automorphism checking [38]. Accordingly, gSpan is not a polynomial-delay method. However, for sparse graphs like chemical compounds, it can scale up to tens of thousands of graphs. As in itemset mining, gSpan might not be feasible when graphs are dense with many edges and similar to each other.

16.3 Graph Classification

Graph classification tasks can either be unsupervised or supervised. Unsupervised methods classify graphs into a certain number of categories by similarity [34, 35]. In supervised classification, a classification rule of graphs is constructed by learning from training data so that novel graphs can be classified with high accuracy.

In training data, we have pairs of labeled graphs (e.g., chemical compounds) and their target values (e.g., biochemical activity). Techinically, supervised methods are more fundamental, because unsupervised methods can be designed from supervised method via probabilistic modeling of latent class labels [34].

The simplest way to apply such pattern mining techniques to graph classification is to build a binary feature vector based on the presence or absence of frequent patterns and apply an off-the-shelf classifier. Such methods are employed in a few chemoinformatics papers [14, 17]. However, they are obviously suboptimal because frequent patterns are not necessarily useful for classification. In chemical data, ubiquitous patterns like C–C or C–C–C are frequent, but have almost no significance.

To discuss pattern mining strategies for graph classification, let us first define the binary classification problem. The task is to learn a prediction rule from the training examples $\{(G_i, y_i)\}_{i=1}^n$, where G_i is a training graph and $y_i \in \{+1, -1\}$ is the associated class label. Let $\mathscr{P}$ be the set of all patterns, i.e., the set of all subgraphs included in at least one training graph, and $d := |\mathscr{P}|$. Then, each graph G_i is encoded as a d-dimensional vector

$$x_{i,p} = \begin{cases} 1 & \text{if } p \subseteq G_i, \\ -1 & \text{otherwise,} \end{cases}$$

This feature space is illustrated in Fig. 16.5.

Since the whole feature space is intractably large, we need to obtain a set of informative patterns without enumerating all patterns (i.e., discriminative pattern mining). This problem is close to feature selection in machine learning, but the difference is that it is not allowed to scan all features. As in feature selection, we can consider the following three categories in discriminative pattern mining methods: filter, wrapper and embedded [18]. In filter methods, discriminative patterns are collected by a mining call before the learning algorithm is started. They employ a simple statistical criterion such as information gain [23, 37]. In wrapper and embedded methods, the learning algorithm chooses features via minimization of a sparsity-inducing objective function. Typically, they have a high dimensional weight vector and most of these weights coverge to zero after optimization. In most cases, the sparsity is induced by L1-norm regularization [30]. The difference between wrapper and embedded methods are subtle, but wrapper methods tend to base on

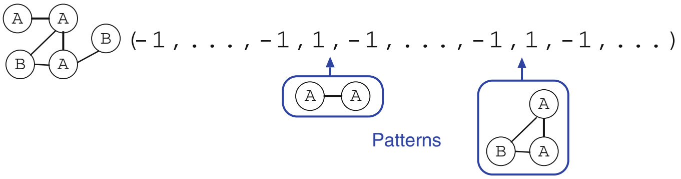


Fig. 16.5 Feature space based on subgraph patterns. The feature vector consists of binary pattern indicators

heuristic ideas by reducing the features recursively (recursive feature elimination) [11]. Graph boosting is an embedded method, but to deal with graphs, we need to combine L1-norm regularization with graph mining.

16.3.1 Formulation of Graph Boosting

The name 'boosting' comes from the fact that linear program boosting (LPBoost) is used as a fundamental computational framework. In chemoinformatics experiments in [30], it was shown that the accuracy of graph boosting is better than walk-based graph kernels [16]. At the same time, key substructures are explicitly discovered.

Our prediction rule is a convex combination of binary indicators $x_{i,j}$, and has the form

$$f(\boldsymbol{x}_i) = \sum_{p \in \mathscr{P}} \beta_p \boldsymbol{x}_{i,p}, \tag{16.1}$$

where $\boldsymbol{\beta}$ is a $|\mathscr{P}|$-dimensional column vector such that $\sum_{p \in \mathscr{P}} \beta_p = 1$ and $\beta_p \geq 0$.

This is a linear discriminant function in an intractably large dimensional space. To obtain an interpretable rule, we need to obtain a *sparse* weight vector $\boldsymbol{\beta}$, where only a few weights are nonzero. In the following, we will present a linear programming approach for efficiently capturing such patterns. Our formulation is based on that of LPBoost [6], and the learning problem is represented as

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\|_1 + \lambda \sum_{i=1}^{n} [1 - \boldsymbol{y}_i f(\boldsymbol{x}_i)]_+ , \tag{16.2}$$

where $\|x\|_1 = \sum_{i=1}^{n} |\boldsymbol{x}_i|$ denotes the ℓ_1 norm of $\boldsymbol{x}$, λ is a regularization parameter, and the subscript "+" indicates positive part. A soft-margin formulation of the above problem exists [6], and can be written as

$$\min_{\boldsymbol{\beta},\boldsymbol{\xi},\rho} -\rho + \lambda \sum_{i=1}^{n} \xi_i \tag{16.3}$$

$$\text{s.t.} \quad \boldsymbol{y}^\top \boldsymbol{X} \boldsymbol{\beta} + \xi_i \geq \rho, \quad \xi_i \geq 0, \quad i = 1, \ldots, n \tag{16.4}$$

$$\sum_{p \in \mathscr{P}} \beta_p = 1, \quad \beta_p \geq 0,$$

where $\boldsymbol{\xi}$ are slack variables, ρ is the margin separating negative examples from positives, $\lambda = \frac{1}{\nu n}$, $\nu \in (0, 1)$ is a parameter controlling the cost of misclassification which has to be found using model selection techniques, such as cross-validation. It is known that the optimal solution has the following ν-property:

Theorem 16.1 ([27]). *Assume that the solution of (16.3) satisfies $\rho \geq 0$. The following statements hold:*

1. *ν is an upperbound of the fraction of* margin errors*, i.e., the examples with*

$$\boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{\beta} < \rho.$$

2. *ν is a lowerbound of the fraction of the examples such that*

$$\boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{\beta} < \rho.$$

Directly solving this optimization problem is intractable due to the large number of variables in $\boldsymbol{\beta}$. So we solve the following *equivalent* dual problem instead.

$$\min_{\boldsymbol{u},v} v \tag{16.5}$$

$$\text{s.t. } \sum_{i=1}^{n} u_i y_i x_{i,p} \le v, \ \forall p \in \mathscr{P} \tag{16.6}$$

$$\sum_{i=1}^{n} u_i = 1, \quad 0 \le u_i \le \lambda, \ i = 1,\dots,n.$$

After solving the dual problem, the primal solution $\boldsymbol{\beta}$ is obtained from the Lagrange multipliers [6]. The dual problem has a limited number of variables, but a huge number of constraints. Such a linear program can be solved by the *column generation* technique [20]: Starting with an empty pattern set, the pattern whose corresponding constraint is violated the most is identified and added iteratively. Each time a pattern is added, the optimal solution is updated by solving the restricted dual problem. Denote by $\boldsymbol{u}^{(k)}, v^{(k)}$ the optimal solution of the restricted problem at iteration $k = 0, 1, \dots,$ and denote by $\hat{\boldsymbol{X}}^{(k)} \subseteq \mathscr{P}$ the set at iteration k. Initially, $\hat{\boldsymbol{X}}^{(0)}$ is empty and $u_i^{(0)} = 1/n$. The restricted problem is defined by replacing the set of constraints (16.6) with

$$\sum_{i=1}^{n} u_i^{(k)} y_i x_{i,p} \le v, \ \forall p \in \hat{\boldsymbol{X}}^{(k)}.$$

The left hand side of the inequality is called as *gain* in boosting literature. After solving the problem, $\hat{\boldsymbol{X}}^{(k)}$ is updated to $\hat{\boldsymbol{X}}^{(k+1)}$ by adding a column. Several criteria have been proposed to select the new columns [8], but we adopt the most simple rule that is amenable to graph mining: We select the constraint with the largest gain.

$$p^* = \operatorname*{argmax}_{p \in \mathscr{P}} \sum_{i=1}^{n} u_i^{(k)} y_i x_{i,p}. \tag{16.7}$$

The solution set is updated as $\hat{\boldsymbol{X}}^{(k+1)} \leftarrow \hat{\boldsymbol{X}}^{(k)} \cup \boldsymbol{X}_{j*}$. In the next section, we discuss how to efficiently find the largest gain in detail.

One of the big advantages of our method is that we have a stopping criterion that guarantees that the optimal solution is found: If there is no $p \in \mathscr{P}$ such that

$$\sum_{i=1}^{n} u_i^{(k)} y_i x_{i,p} > v^{(k)}, \tag{16.8}$$

then the current solution is the optimal dual solution. Empirically, the patterns found in the last few iterations have negligibly small weights. The number of iterations can be decreased by relaxing the condition as

$$\sum_{i=1}^{n} u_i^{(k)} y_i x_{i,p} > v^{(k)} + \epsilon, \tag{16.9}$$

Let us define the primal objective function as $V = -\rho + \lambda \sum_{i=1}^{n} \xi_i$. Due to the convex duality, we can guarantee that, for the solution obtained from the early termination (16.9), the objective satisfies $V \leq V^* + \epsilon$, where V^* is the optimal value with the exact termination (16.8) [6].

16.3.2 Optimal Pattern Search

As mentioned in (16.7), our aim is to find the optimal hypothesis that maximizes the gain $g(p)$.

$$g(p) = \sum_{i=1}^{n} u_i^{(k)} y_i x_{i,p}. \tag{16.10}$$

For efficient search, it is important to minimize the size of the actual search space. To this aim, *tree pruning* is crucially important: Suppose the search tree is generated up to the pattern p and denote by g^* the maximum gain among the ones observed so far. If it is guaranteed that the gain of any supergraph p' is not larger than g^*, we can avoid the generation of downstream nodes without losing the optimal pattern. For gain maximization, we employ the following pruning condition.

Theorem 16.2. *[19, 22] Let us define*

$$\mu(p) = 2 \sum_{\{i \mid y_i = +1, p \subseteq G_i\}} u_i^{(k)} - \sum_{i=1}^{n} y_i u_i^{(k)}.$$

If the following condition is satisfied,

$$g^* > \mu(p), \tag{16.11}$$

the inequality $g(p') < g^*$ holds for any p' such that $p \subseteq p'$.

Algorithm 2 gBoost algorithm: main part

```
$\hat{\boldsymbol{X}}^{(0)} = \emptyset, \boldsymbol{u}_i^{(0)} = 1/n, k = 0$
loop
    Find the optimal pattern $p^*$ based on $\boldsymbol{u}^{(k)}$    ▷ Algorithm 2
    if termination condition (16.9) holds then
        break
    end if
    $\hat{\boldsymbol{X}} \leftarrow \hat{\boldsymbol{X}} \cup \boldsymbol{X}_{j^*}$
    Solve the restricted dual problem (16.5) to obtain $\boldsymbol{u}^{(k+1)}$
    $k = k + 1$
end loop
```

Algorithm 3 Finding the optimal pattern

```
procedure OPTIMAL PATTERN
    Global variables: $g^*$, $p^*$
    $g^* = -\infty$
    for $p \in$ DFS codes with single nodes do
        project($p$)
    end for
    return $p^*$
end procedure
function PROJECT($p$)
    if $p$ is not a minimum DFS code then
        return
    end if
    if pruning condition (16.11) holds then    ▷ Theorem 2
        return
    end if
    if $g(p) > g^*$ then
        $g^* = g(p), p^* = p$
    end if
    for $p' \in$ rightmost extensions of $p$ do
        project($p'$)
    end for
end function
```

The gBoost algorithm is summarized in Algorithms 2 and 3.

16.3.3 Computational Experiments

In [30], it is shown that graph boosting performs better than graph kernels in classification accuracy in chemical compound datasets. The top 20 discriminative subgraphs for a mutagenicity dataset called CPDB are displayed in Fig. 16.6. We found that the top three substructures with positive weights (0.0672, 0.0656, 0.0577) correspond to known *toxicophores* [17]. They correspond to *aromatic amine*, *aliphatic halide*, and *three-membered heterocycle*, respectively. In addition,

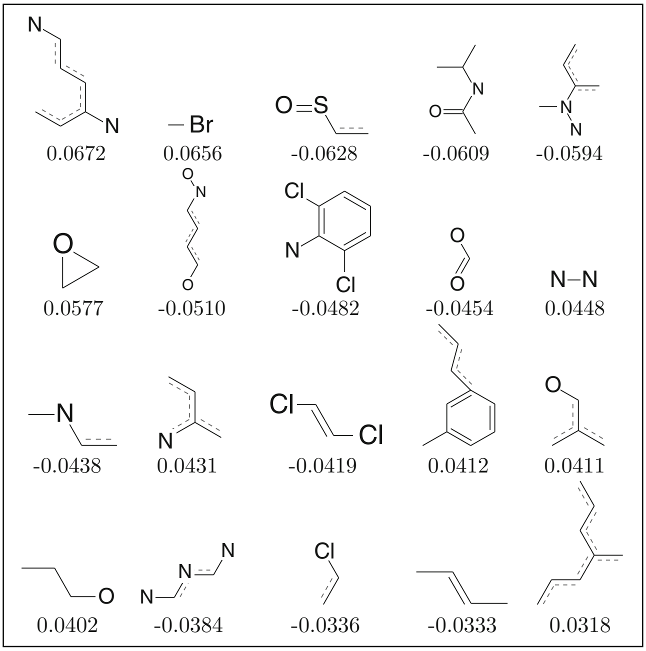


Fig. 16.6 Top 20 discriminative subgraphs from the CPDB dataset. Each subgraph is shown with the corresponding weight, and ordered by the absolute value from the top left to the bottom right. H atom is omitted, and C atom is represented as a dot for simplicity. Aromatic bonds appeared in an open form are displayed by the combination of *dashed* and *solid lines*

the patterns with weights 0.0431, 0.0412, 0.0411 and 0.0318 seem to be related to *polycyclic aromatic systems*. Only from this result, we cannot conclude that graph boosting is better in general data. However, since important chemical substructures cannot be represented in paths, it would be reasonable to say that subgraph features are better in chemical data.

16.3.4 Related Methods

Graph algorithms can be designed based on existing statistical frameworks (i.e., mother algorithms). It allows us to use theoretical results and insights accumulated in the past studies. In graph boosting, we employed LPboost as a mother algorithm. It is possible to employ other algorithms such as partial least squares regression (PLS) [29] and least angle regression (LARS) [33].

When applied to ordinary vectorial data, partial least squares regression extracts a few orthogonal features and perform least squares regression in the projected space [28]. A PLS feature is a linear combination of original features, and it is often the case that correlated features are summarized into a PLS feature. Sometimes, the subgraph features chosen by graph boosting is not robust against bootstrapping or other data perturbations, whereas the classification accuracy is quite stable. It is due to strong correlation among features corresponding to similar subgraphs. The graph mining version of PLS, gPLS [29], solves this problem by summarizing similar subgraphs into each feature (Fig. 16.7). Since only one graph mining call is required to construct each feature, gPLS can build the classification rule more quickly than graph boosting.

In graph boosting, it is necessary to set the regularization parameter λ in (16.2). Typically it is determined by cross validation, but there is a different approach called "regularization path tracking". When $\lambda = 0$, the weight vector converges to the origin. As λ is increased continuously, the weight vector draws a piecewise linear path. Because of this property, one can track the whole path by repeating to jump to the next turning point. We combined the tracking with graph mining in [33]. In ordinary tracking, a feature is added or removed at each turning point. In our graph version, a subgraph to add or remove is found by a customized gSpan search.

The examples shown above were for supervised classification. For unsupervised clustering of graphs, the combinations with the EM algorithm [34] and the Dirichlet process [35] have been reported.

16.3.5 New Applications of Graph Mining

A main advantageous point of using graph mining rather than graph kernels is that the subgraphs that are correlated with class labels can be detected, which should be appreciated in many application domains. Traditionally, graph mining methods are mainly used for small chemical compounds [7, 21]. However, new application areas are emerging. One is image processing [25], where geometric relationship between points is represented as edges. Bug detection is an interesting area, where the relationships of APIs are represented as directed graphs and anomalous patterns are detected to identify bugs [5, 9]. Also natural language processing is an attractive area, where the relationships between words are represented as a graph (e.g., predicate-argument structures) and key phrases are identified as subgraphs [19].

16.4 Concluding Remarks

As mentioned briefly in Sect. 16.1, there are two different methods for graph data processing: graph kernels and graph mining. The graph kernel is a similarity measure between two graphs. On the other hand, graph mining methods can derive characteristic subgraphs that can be used for any subsequent machine learning algorithms. I have the impression that graph kernels are more frequently applied so far.

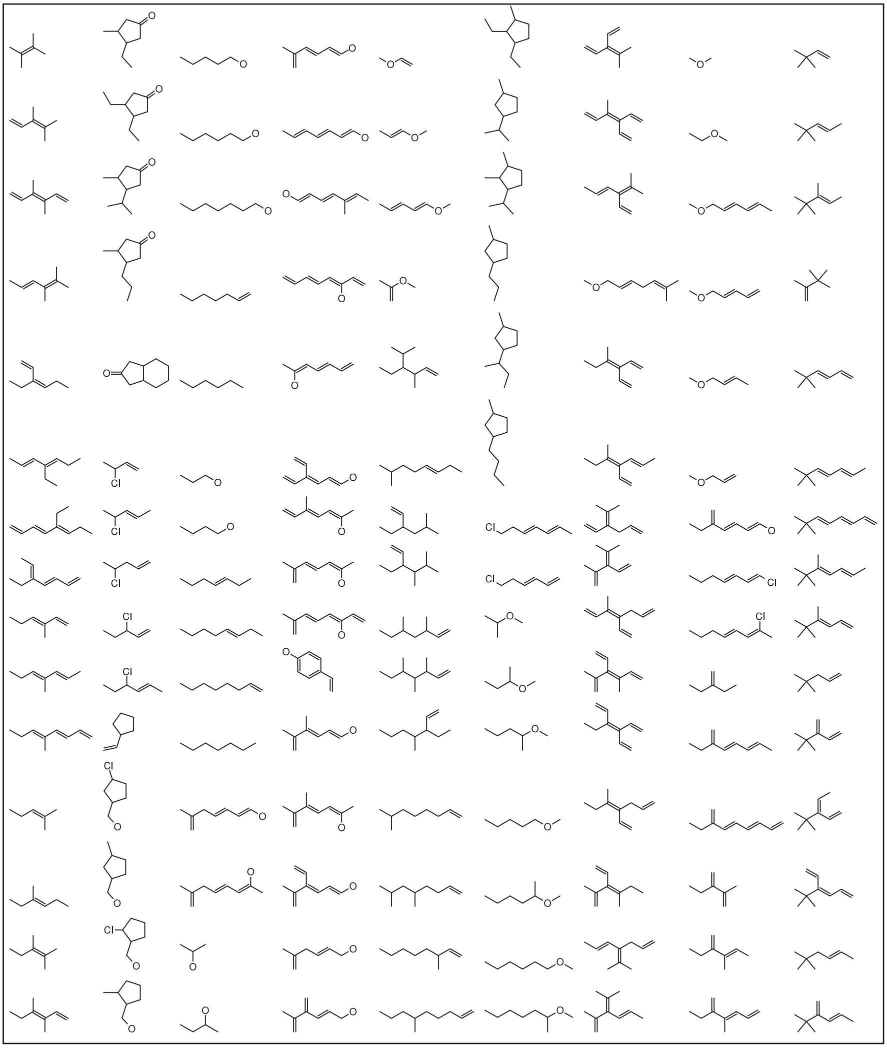


Fig. 16.7 Patterns obtained by gPLS. Each column corresponds to the patterns of a PLS component.

Probably it is due to the fact that graph kernels are easier to implement and currently used graph datasets are not so large. However, graph kernels are not suitable for very large data, because it takes $O(n^2)$ time to derive the kernel matrix of n training graphs, which is very hard to improve. Toward large scale data, graph mining methods seem more promising because it takes only $O(n)$ time. Nevertheless, there remains much to be done in graph mining methods. Existing methods such as gSpan enumerate all subgraphs satisfying a certain frequency-based criterion. However, it is often pointed out that, for graph classification, it is not always necessary to enumerate all subgraphs. Recently, Boley and Grosskreutz proposed a uniform sampling method of frequent itemsets [3]. Such theoretically guaranteed sampling procedures will certainly contribute to graph classification as well.

One fact that hinders the dissemination of graph mining methods is that it is not common to make the code public in machine learning and data mining community. We have made several easy-to-use graph mining codes available in the gBoost package (www.nowozin.net/sebastian/gboost/).

Whenever there are multiple interacting elements, graph representation comes into consideration. Graph-data analysis would be necessary in virtually all fields of computer science and graph classification will certainly play a role there.

References

1. Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. In *Proceedings of VLDB 1994* (pp. 487–499).
2. Asai, T., Abe, K., Kawasoe, S., Arimura, H., Sakamoto, H., & Arikawa, S. (2002). Efficient substructure discovery from large semi-structured data. In *Proceedings of 2nd SIAM data mining conference (SDM)* (pp. 158–174).
3. Boley, M., & Grosskreutz, H. (2008). A randomized approach for approximating the number of frequent sets. In *Proceedings of the 8th IEEE international conference on data mining* (pp. 43–52).
4. Borgwardt, K. M., Ong, C. S., Schönauer, S., Vishwanathan, S. V. N., Smola, A. J., & Kriegel, H.-P. (2006). Protein function prediction via graph kernels. *Bioinformatics*, *21*(Suppl. 1), i47–i56.
5. Cheng, H., Lo, D., Zhou, Y., Wang, X., & Yan, X. (2009). Identifying bug signatures using discriminative graph mining. In *Proceedings of the 18th international symposium on software testing and analysis* (pp. 141–152).
6. Demiriz, A., Bennet, K. P., & Shawe-Taylor, J. (2002). Linear programming boosting via column generation. *Machine Learning*, *46*(1–3), 225–254.
7. Deshpande, M., Kuramochi, M., Wale, N., & Karypis, G. (2005). Frequent sub-structure-based approaches for classifying chemical compounds. *IEEE Transactions on Knowledge and Data Engineering*, *17*(8), 1036–1050.
8. du Merle, O., Villeneuve, D., Desrosiers, J., & Hansen, P. (1999). Stabilized column generation. *Discrete Mathematics*, *194*, 229–237.
9. Eichinger, F., Böhm, K., & Huber, M. (2008). Mining edge-weighted call graphs to localise software bugs. In *Proceedings of the European conference on machine learning and principles and practice of knowledge discovery in databases (ECML PKDD)* (pp. 333–348).
10. Gasteiger, J., & Engel, T. (2003). *Chemoinformatics: A textbook*. Weinheim, Germany: Wiley-VCH.
11. Guyon, I., Weston, J., Bahnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*(1–3), 389–422.
12. Hamada, M., Tsuda, K., Kudo, T., Kin, T., & Asai, K. (2006). Mining frequent stem patterns from unaligned RNA sequences. *Bioinformatics*, 22, 2480–2487.
13. Han, J., & Kamber, M. (2000). *Data mining: Concepts and techniques*. San Francisco: Morgan Kaufmann.
14. Helma, C., Cramer, T., Kramer, S., & Raedt, L. D. (2004). Data mining and machine learning techniques for the identification of mutagenicity inducing substructures and structure activity relationships of noncongeneric compounds. *Journal of Chemical Information Computer Science*, *44*, 1402–1411.

15. Inokuchi, A. (2005). Mining generalized substructures from a set of labeled graphs. In *Proceedings of the 4th IEEE internatinal conference on data mining* (pp. 415–418). Los Alamitos, CA: IEEE Computer Society.
16. Kashima, H., Tsuda, K., & Inokuchi, A. (2003). Marginalized kernels between labeled graphs. In *Proceedings of the 21st international conference on machine learning* (pp. 321–328). New York: AAAI.
17. Kazius, J., Nijssen, S., Kok, J., Bäck, T., & Ijzerman, A. P. (2006). Substructure mining using elaborate chemical representation. *Journal of Chemical Information Modeling*, *46*, 597–605.
18. Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *1–2*, 273–324.
19. Kudo, T., Maeda, E., & Matsumoto, Y. (2005). An application of boosting to graph classification. In *Advances in neural information processing systems* (Vol. 17, pp. 729–736). Cambridge, MA: MIT.
20. Luenberger, D. G. (1969). *Optimization by vector space methods*. New York: Wiley.
21. Mahé, P., Ueda, N., Akutsu, T., Perret, J.-L., & Vert, J.-P. (2005). Graph kernels for molecular structure – activity relationship analysis with support vector machines. *Journal of Chemical and Information Modeling*, *45*, 939–951.
22. Morishita, S. (2001). Computing optimal hypotheses efficiently for boosting. In *Discovery science* (pp. 471–481).
23. Morishita, S., & Sese, J. (2000). Traversing itemset lattices with statistical metric pruning. In *Proceedings of ACM SIGACT-SIGMOD-SIGART symposium on database systems (PODS)* (pp. 226–236).
24. Nijssen, S., & Kok, J. N. (2004). A quickstart in frequent structure mining can make a difference. In *Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 647–652). New York: ACM Press.
25. Nowozin, S., Tsuda, K., Uno, T., Kudo, T., & Bakir, G. (2007). Weighted substructure mining for image analysis. In *IEEE computer society conference on computer vision and pattern recognition (CVPR)*. Los Alamitos, CA: IEEE Computer Society.
26. Pei, J., Han, J., Mortazavi-asl, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., & Hsu, M. (2004). Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Transactions on Knowledge and Data Engineering*, *16*(11), 1424–1440.
27. Rätsch, G., Mika, S., Schölkopf, B., & Müller, K.-R. (2002). Constructing boosting algorithms from SVMs: An application to one-class classification. *IEEE Transactions on Pattern Analysis Machine Intelligence*, *24*(9), 1184–1199.
28. Rosipal, R., & Krämer, N. (2006). Overview and recent advances in partial least squares. In *Subspace, latent structure and feature selection techniques* (pp. 34–51). Springer.
29. Saigo, H., Krämer, N., & Tsuda, K. (2008). Partial least squares regression for graph mining. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 578–586).
30. Saigo, H., Nowozin, S., Kadowaki, T., Kudo, T., & Tsuda, K. (2008). GBoost: A mathematical programming approach to graph classification and regression. *Machine Learning*.
31. Sanfeliu, A., & Fu, K. S. (1983). A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on System, Man and Cybernetics*, *13*, 353–362.
32. Schölkopf, B., & Smola, A. J. (2002). *Learning with Kernels: Support vector machines, regularization, optimization, and beyond*. Cambridge, MA: MIT.
33. Tsuda, K. (2007). Entire regularization paths for graph data. In *Proceedings of the 24th international conference on machine learning* (pp. 919–926).
34. Tsuda, K., & Kudo, T. (2006). Clustering graphs by weighted substructure mining. In *Proceedings of the 23rd international conference on machine learning* (pp. 953–960). New York: ACM.
35. Tsuda, K., & Kurihara, K. (2008). Graph mining with variational dirichlet process mixture models. In *SIAM Conference on Data Mining (SDM)*.

36. Wale, N., & Karypis, G. (2006). Comparison of descriptor spaces for chemical compound retrieval and classification. In *Proceedings of the 2006 IEEE international conference on data mining* (pp. 678–689).
37. Yan, X., Cheng, H., Han, J., & Yu, P. S. (2008). Mining significant graph patterns by leap search. In *Proceedings of the ACM SIGMOD international conference on management of data* (pp. 433–444).
38. Yan, X., & Han, J. (2002). gSpan: Graph-based substructure pattern mining. In *Proceedings of the 2002 IEEE international conference on data mining* (pp. 721–724). Los Alamitos, CA: IEEE Computer Society.
39. Zaki, M., Parthasarathy, S., Ogihara, M., & Li, W. (1997). New algorithms for fast discovery of association rules. In *KDD 1997* (pp. 283–286).

Chapter 17
Hidden Markov Random Field Models for Network-Based Analysis of Genomic Data

Hongzhe Li

Abstract Graphs and networks are common ways of depicting biological information. In biology, many different biological processes are represented by graphs, such as regulatory networks, metabolic pathways and protein-protein interaction networks. This kind of a priori use of graphs is a useful supplement to the standard numerical data such as microarray gene expression data and single nucleotide polymorphisms (SNPs) data. How to incorporate such a prior network information into analysis of numerical data raises interesting statistical problems. Representing the genetic networks as undirected graphs, we have developed several approaches for identifying differentially expressed genes and genes or SNPs associated with diseases in a unified framework of hidden Markov random field (HMRF) models. Different from the traditional empirical Bayes approaches for analysis of gene expression data, the HMRF-based models account for the prior dependency among the genes on the network and therefore effectively utilize the prior network information in identifying the subnetworks of genes that are perturbed by experimental conditions. In this paper, we briefly review the basic setup of the HMRF models and the emission probability functions for some problems often encountered in analysis of microarray gene expression and SNPs data. We also present some interesting areas that require further research.

17.1 Introduction

Microarray gene expression studies have been widely used in biomedical research. The most common problem is to identify genes that are perturbed by experimental conditions or genes that areassociated with certain covariates or outcomes. Empirical Bayes-based methods are one of the most popular statistical approaches for analysis of microarray gene expression data in order to account for the parallel

H. Li
University of Pennsylvania
e-mail: hongzhe@upenn.edu

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_17,

© Springer-Verlag Berlin Heidelberg 2011

nature of the inference in microarrays and to borrow information from the ensemble of genes that can enhance the inference about each gene individually. Efron et al. [8] used a non-parametric empirical Bayes approach for analyzing the factorial microarray gene expression data. Lonnstedt and Speed [17] took a parametric empirical Bayes approach using a simple mixture of normal models and a conjugate prior and derived the closed-formed posterior odds of differential expression for each gene. Smyth [22] developed the hierarchical model of Lonnstedt and Speed [17] into a practical approach for general microarray experiments in the framework of linear models with arbitrary coefficients and contrasts of interests. Smyth [22] also derived the posterior odds statistic in terms of a moderated t-statistic in which posterior residual standard deviations are used in place of ordinary standard deviations. Yuan and Kendzioski [32], Tai and Speed [24] and Hong and Li [9] developed different empirical Bayes methods for identifying the temporally differentially expressed genes based on time course gene expression data.

While these empirical Bayes methods have proved to be very useful for identifying the differentially expressed genes or genes that are related to certain covariates, they make a key assumption that genes are independent. However, since many biological processes are involved in activation of multiple pathways of correlated genes, the genes with regulatory relationships are expected to be dependent. These dependent genes often interact with each other to form molecular modules that affect the cellular and clinical phenotypes [10]. One approach to modeling the dependency among the genes and to identifying the molecular modules is to utilize the prior genetic regulatory network information. Information about gene regulatory dependence has been accumulated from many years of biomedical experiments and is summarized in the form of pathways and networks and assembled into pathway databases. Some well-known pathway databases include KEGG (http://www.genome.jp/kegg/) [12], BioCarta (www.biocarta.com), BioCyc (www.biocyc.org) and human protein-protein interaction networks HPRD [20], BIND (www.bind.ca) [4]. As an example, Fig. 17.1 shows the KEGG human regulatory network [12], consisting of 33 interconnected regulatory pathways. Such prior network information was shown to be very useful in interpreting gene expression data by improving sample classification and improving detection of differentially expressed genes, especially when the sample sizes are small. There has been a great interest in developing statistical and computational methods that can integrate the prior biological network information into the analysis of genomic data, especially into the analysis of microarray gene expression data (see Ideker and Sharan [10] for a review). However, these methods were mainly developed from computational aspects without formal statistical modeling [25, 26] and focused on using the network information to enhance detection of modules of co-expressed genes [27].

Representing the known genetic regulatory network as an undirected graph, Wei and Li [29, 30] and Wei and Pan [28] have recently developed hidden Markov random field (HMRF)-based models for identifying the subnetworks that show differential expression patterns between two conditions, and have demonstrated using both simulations and applications to real data sets that the procedure is more sensitive in identifying the differentially expressed genes than those procedures that do

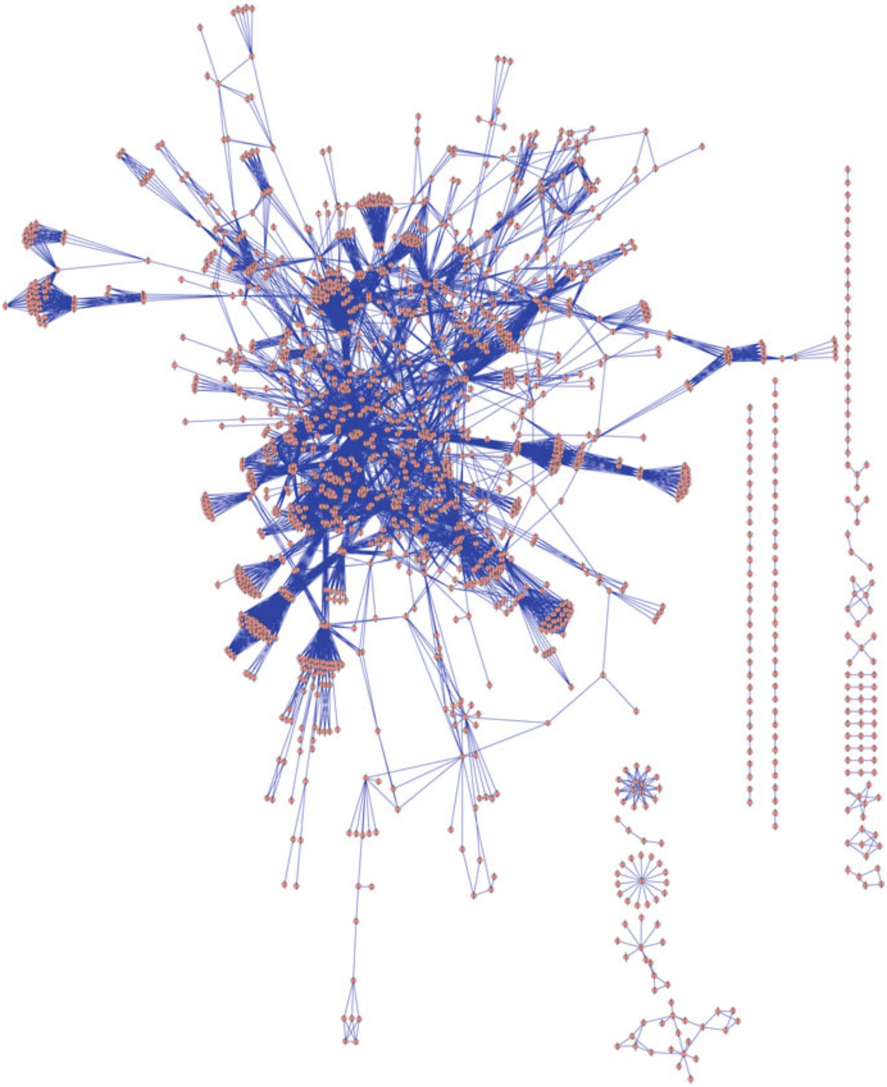


Fig. 17.1 Undirected graph of the KEGG regulatory network, consisting of 33 interconnected regulatory pathways. There are a total of 1,663 genes (*nodes*) and 8,011 regulatory relationships (*edges*)

not utilize pathway structure information. The HMRF models were further extended for analysis of microarray time course gene expression data [30] and more general multivariate gene expression data [31]. They were also extended for general linear models for microarray gene expression data ([15]) and analysis of genetic association data ([16]). MRF models have also been applied to network-based prediction of protein function [6, 7, 21].

In this paper, we review the general HMRF modeling framework for network-based analysis of microarray gene expression data and analysis of single nucleotide polymorphism data in genetic association studies. This review summarizes the methods presented in the following papers: Wei and Li [29, 30], Wei et al. [31], Li et al. [15] and Li et al. [16]. We briefly outline the HMRF formulation for several problems in genomic data analysis and summarize the iterative conditional modes algorithm for parameter estimation. Finally, we outline several other interesting problems in genomics that require further methodological development.

17.2 Networks, Graphs and Markov Random Field Models

Suppose that we have a network of known pathways that can be represented as an undirected graph $G = (V, E)$, where V is the set of nodes that represent genes or proteins coded by genes and E is the set of edges linking two genes with a regulatory relationship or a link in protein protein interaction network. Let $p = |V|$ be the number of genes that this network contains. Note the gene set V is often a subset of all the genes that are probed on the gene expression arrays. If we want to include all the genes that are probed on the expression arrays, we can expand the network graph G to include isolated nodes, which are those genes that are probed on the arrays but are not part of the known biological network. For two genes g and g', if they are linked on the network, we write $g \sim g'$. For a given gene g, let $N_g = \{g' : g \sim g' \in E\}$ be the set of genes that are linked to gene g and $m_g = |N_g|$ be the degree for gene g.

Let z_g be an indicator variable that defines whether the gth gene is perturbed by certain experimental condition or associated with some covariate. Its definition is problem-specific and will be clear in the next several sections. The key to the HMRF approach to network-based analysis of genomic data is that instead of assuming that $z_1, \dots, z_p$ are independently and identically distributed Bernoulli random variables, we assume that they are dependent on the network, whose dependency can be modeled as a simple discrete Markov random field. Specifically, Wei and Li [29, 30] proposed to model the dependency of $z = (z_1, \dots, z_g, \dots, z_p)^T$ using a discrete Markov random field model with the following distribution:

$$p(z; \Phi) \propto \exp(\gamma \sum_{g=1}^{p} z_g + \eta \sum_{g \sim g'} I\{z_g = z_{g'}\}), \qquad (17.1)$$

where $\Phi = (\gamma, \eta)$, γ is related to the marginal probability of association and η measures the pair-wise dependency. We require η to be non-negative to discourage neighboring genes with different states. Given the states of all other genes, the conditional probability of gene i with state z_g can be easily derived as

$$p_g(z_g | z_{\partial_g}; \Phi) \propto \exp(\gamma z_g + \eta \mu_g(z_g)), \qquad (17.2)$$

where z_{∂_g} represents the neighbors of gene g and $u_g(z_g)$ denotes the number of neighbors of gene g having state z_g [2,3]. In order to account for different degrees of the nodes (i.e., different numbers of neighboring genes on the network), we propose to modify the conditional probability (17.2) as

$$p_g(z_g|z_{\partial_g};\Phi) \propto \exp(\gamma z_g - \eta\mu_g(1-z_g)/m_g),$$

where m_g is the number of neighbors of the gth gene. Different from the conditional probability in Eq. 17.2, this modified conditional probability does not seem to correspond to a well-defined joint distribution of the $z_g, g = 1, \ldots, p$. This conditional probability should be used for account for different numbers of the neighbors of genes on the network. Finally, we assume that the true state z^* is a realization of a discrete MRF with a specified distribution $p(z)$ defined by Eq. 17.1. The goal of analysis is to infer the true state z^* based on the data observed.

17.3 HMRF Models for Network-Based Analysis of Gene Expression Data

In this section, we make the definition of z_g for the gth gene explicit and define the probability models to relate z_g to the observed data $\mathbf{O}_g$ for various problems in analysis of microarray gene expression data. We can treat the data observed on the network $\mathbf{O} = \{\mathbf{O}_g, g \in V\}$ as an observable random field. Let $f(\mathbf{O}_g|z_g)$ be the emission probability function, which needs to be specified differently for different problems. Due to small sample sizes in typical microarray experiments, emission probability functions derived from hierarchical models and empirical Bayes methods are often preferred. We review in the following some key empirical Bayes methods published in literature that we used to derive the emission probability functions for our HMRF models.

17.3.1 Identification of Differentially Expressed Modules

Wei and Li [29] was the first to propose the use of HMRF for identifying the differentially expressed genes between two experimental conditions using the network structure information. Consider the simple problem of identifying the differentially expressed genes between two experimental conditions. Let $\mathbf{Y_g} = (y_{g1}, y_{g2}, \ldots, y_{gm}; y_{g(m+1)}, \ldots, y_{g(m+n)})$ be the observed mRNA expression level of gene g across $m + n$ samples, where the first m samples are from condition 1 and the next n samples are from condition 2. We are interested in testing

$$H_{g0} : \mu_{1g} = \mu_{2g},$$

where μ_{kg} is the mean expression level of the gth gene under condition k. Assume that the gth gene can have two states, labeled 0 and 1, representing equally expression (EE) and differential expression (DE), respectively, i.e.,

$$z_g = \begin{cases} 1 \text{ if } \mu_{1g} \neq \mu_{2g} \text{ (gene } g \text{ is DE)} \\ 0 \text{ if } \mu_{1g} = \mu_{2g} \text{ (gene } g \text{ is EE).} \end{cases}$$

Under the Gamma-Gamma hierarchical models for gene expression data ([13,19, 29]), one can show that

$$f(\mathbf{Y_g}|z_g = 1) = K_1 K_2 \frac{\left(\prod_{j=1}^{m+n} y_{gj}\right)^{\alpha-1}}{\left(v + y_{g.m}\right)^{m\alpha+\alpha_0}\left(v + y_{g.n}\right)^{n\alpha+\alpha_0}},$$

$$f(\mathbf{Y_g}|z_g = 0) = K \frac{\left(\prod_{j=1}^{m+n} y_{gj}\right)^{\alpha-1}}{\left(v + y_{g.m} + y_{g.n}\right)^{(m+n)\alpha+\alpha_0}},$$

where $y_{g.m} = \sum_{j=1}^{m} y_{gj}$, $y_{g.n} = \sum_{j=m+1}^{m+n} y_{gj}$,

$$K_1 = \frac{v^{\alpha_0}\Gamma(m\alpha + \alpha_0)}{\Gamma^m(\alpha)\Gamma(\alpha_0)},\ K_2 = \frac{v^{\alpha_0}\Gamma(n\alpha + \alpha_0)}{\Gamma^n(\alpha)\Gamma(\alpha_0)},\ \text{and} K = \frac{v^{\alpha_0}\Gamma((m+n)\alpha + \alpha_0)}{\Gamma^{m+n}(\alpha)\Gamma(\alpha_0)}.$$

These two probability distributions specify the emission probability distributions with parameters $\Theta = (\alpha_0, \alpha, v)$, where α is the common shape parameter of the gamma distribution of the gene expression level and α_0 and v are the shape and scale parameters of the gamma prior of the inverse mean expression levels α/μ_g (see Kendziorski et al. [13] and Wei and Li [29] for details of the derivations).

17.3.2 Analysis of Time Course Gene Expression Data

Microarray time course gene expression data are often collected to capture the dynamic nature of gene expression during a given biological process. Wei and Li [29] developed a spatial-temporal HMRF model to identify the DE genes at different time points during the time course, taking into account both the network dependency and the time dependency of the differential expression states. Wei et al. [31] proposed another approach with the goal of identifying the temporally differentially expressed genes. Specifically, consider the multivariate gene expression data measured under two different conditions over k dosage levels or time points, with n independent samples measured under one condition and m independent samples measured under another condition. For each experiment, we assume that the expression levels of p genes are measured. For a given gene g, we denote these data as $i.i.d.$ $k \times 1$ random vectors $\mathbf{Y}_{g1}, \ldots, \mathbf{Y}_{gn}$ for condition 1 and $\mathbf{Z}_{g1}, \ldots, \mathbf{Z}_{gm}$ for condition 2. We further assume that $\mathbf{Y}_{gi} \sim N_k(\boldsymbol{\mu}_{gy}, \Sigma_g)$ and $\mathbf{Z}_{gi} \sim N_k(\boldsymbol{\mu}_{gz}, \Sigma_g)$. For a given gene g, the null hypothesis of interest is

$$H_{g0} : \boldsymbol{\mu}_{gy} = \boldsymbol{\mu}_{gz}.$$

Let $\boldsymbol{\mu}_g = \boldsymbol{\mu}_{gy} - \boldsymbol{\mu}_{gz}$ and define

$$z_g = \begin{cases} 1 \text{ if } \boldsymbol{\mu}_g \neq 0 \\ 0 \text{ if } \boldsymbol{\mu}_g = 0 \end{cases}$$

To link z_g to the observed time-course gene expression data, we need to define the emission probabilities. Following Tai and Speed [24], we take an empirical Bayes approach. Let $\bar{\mathbf{Y}} = (\mathbf{Y}_1 + \cdots + \mathbf{Y}_n)/n, \bar{\mathbf{Z}} = (\mathbf{Z}_1 + \cdots + \mathbf{Z}_m)/m, \bar{\mathbf{X}} = \bar{\mathbf{Y}} - \bar{\mathbf{Z}}, \mathbf{S}_{\mathbf{y}} = (n-1)^{-1} \sum_{i=1}^{n} (\mathbf{Y}_i - \bar{\mathbf{Y}})(\mathbf{Y}_i - \bar{\mathbf{Y}})', \mathbf{S}_{\mathbf{z}} = (m-1)^{-1} \sum_{i=1}^{m} (\mathbf{Z}_i - \bar{\mathbf{Z}})(\mathbf{Z}_i - \bar{\mathbf{Z}})', \mathbf{S} = (n+m-2)^{-1}((n-1)\mathbf{S}_{\mathbf{y}} + (m-1)\mathbf{S}_{\mathbf{z}})$. Tai and Speed [24] introduced a hierarchical model and showed that $\bar{\mathbf{X}}$ and S are sufficient statistics for testing the null hypothesis (17.3). They further showed that

$$\begin{aligned} f(\bar{\mathbf{X}}, S|I = 1) = {} & \frac{\Gamma_k((N+\nu)/2)}{\Gamma_k((N-1)/2)\Gamma_k(\nu/2)} \\ & \times (N-1)^{\frac{k(N-1)}{2}} \nu^{-\frac{kN}{2}} (\pi(n^{-1} + m^{-1} + \eta^{-1}))^{-\frac{k}{2}} \\ & \times \frac{|\Lambda|^{-\frac{N}{2}} |\mathbf{S}|^{\frac{N-k-2}{2}}}{|\mathbf{I}_k + ((n^{-1} + m^{-1} + \eta^{-1})\nu\Lambda)^{-1}\bar{\mathbf{X}}\bar{\mathbf{X}}' + S*|^{\frac{N+\nu}{2}}}, \end{aligned}$$

and

$$\begin{aligned} f(\bar{\mathbf{X}}, |I = 0) = {} & \frac{\Gamma_k((N+\nu)/2)}{\Gamma_k((N-1)/2)\Gamma_k(\nu/2)} \\ & \times (N-1)^{\frac{k(N-1)}{2}} \nu^{-\frac{kN}{2}} (\pi(n^{-1} + m^{-1}))^{-\frac{k}{2}} \\ & \times \frac{|\Lambda|^{-\frac{N}{2}} |\mathbf{S}|^{\frac{N-k-2}{2}}}{|\mathbf{I}_k + ((n^{-1} + m^{-1})\nu\Lambda)^{-1}\bar{\mathbf{X}}\bar{\mathbf{X}}' + S*|^{\frac{N+\nu}{2}}}, \end{aligned}$$

where $N = n + m - 1$, $S^* = (\nu\Lambda/(N-1))^{-1}\mathbf{S}$, ν and $\nu\Lambda$ are the degrees of freedom and scale matrix in the prior inverse Wishart distribution of Σ_g, and η is a scale parameter (See Tai and Speed [24] for detailed derivations). Thus, given $z_g = 1$, the probability density function of the data is a function of $\bar{\mathbf{X}}$ and $\bar{\mathbf{S}}$ only, which follows a Student-Siegel distribution [1]. Following Aitchison and Dunsmore's and Tai and Speed's notation, this distribution is denoted by $StSi_k(\nu, \mathbf{0}, (n^{-1} + m^{-1} + \eta^{-1})\Lambda, N-1, (N-1)^{-1}\nu\Lambda)$. Similarly, the distribution of $f(\mathbf{X}, S|I = 0)$ follows $StSi_k(\nu, \mathbf{0}, (n^{-1} + m^{-1})\Lambda, N-1, (N-1)^{-1}\nu\Lambda)$. The parameters associated with these two emission probability distributions are $\Theta = (\eta, \nu, \Lambda)$ with a positive definite constraint on the covariance matrix Λ.

17.3.3 Analysis of Gene Expression Data Using General Linear Models

Smyth [22] proposed empirical Bayes methods for general linear models for analysis of microarray gene expression. The methods are very general and can handle general design matrices. Li et al. [15] extended the HMRF model to more general linear models. Assume that we have a set of n microarrays (samples), we want to determine how the experimental conditions affect the expression levels of genes and which genes or subnetworks of genes are affected. Let $Y = (Y_1, \dots, Y_g, \dots, Y_p)$ denote the microarray gene expression profiling data matrix ($n \times p$) of p genes over n samples, where Y_g is the mRNA expression level of gene g for the n samples. Let $X = (x_1, \dots, x_n)^T$ be the $n \times q$ covariate matrix of the n samples, where x_i represents the q-dimensional covariate vector for the ith sample. Depending on designs of the experiments, this vector could correspond to the design matrix or other general covariates (see [22]) for specification of the design matrices for various microarray experiments). We assume the following linear model for gene expression level for the gth gene:

$$\begin{aligned} Y_g &= \mu_g + X\alpha_g + \epsilon_g, \\ var(\epsilon_g) &= \sigma_g^2\mathbf{I}, g = 1, \dots, p, \end{aligned} \tag{17.3}$$

where α_g a coefficient vector and ϵ_g is the vector of random errors. Let $\hat{\alpha}_g$ be the least squares estimate of α_g and $\hat{\sigma}_g^2$ be the estimate of σ_g^2 based on this model. Further let $Var(\hat{\alpha}_g) = V_g\hat{\sigma}_g^2$ be the estimated covariance matrix, where V_g is a positive definite matrix based on the design matrix X.

Certain contrasts of the coefficients are assumed to be of biological interest and these are defined by $\beta_g = C^T\alpha_g$, where C is a contrast vector. The β_g can then be estimated by $\hat{\beta}_g = C^T\hat{\alpha}_g$ with its variance estimated by $Var(\hat{\beta}_g) = C^T V_g C\hat{\sigma}_g^2 = v_g\hat{\sigma}_g^2$. Based on model (17.3), we have

$$\begin{aligned} \hat{\beta}_g|\beta_g, \sigma_g^2 &\sim N(\beta_g, v_g\sigma_g^2), \\ \hat{\sigma_g}^2|\sigma_g^2 &\sim \frac{\sigma_g^2}{d_g}\chi^2_{d_g}, \end{aligned}$$

where $d_g = n - q$ is the residual degrees of freedom.

We are interested in testing whether individual contrast value β_g to be zero. To achieve this goal, we introduce a random vector $z = (z_1, \dots, z_g, \dots, z_p)^T$, representing the gene states, where

$$z_g = \begin{cases} 1 \text{ if } \beta_g \neq 0 \\ 0 \text{ if } \beta_g = 0. \end{cases}$$

Assuming an inverse Chi-square prior for the σ_g^2 with means s_0^2 and degrees of d_0, and a normal prior for β_g when $z_g = 1$ with a scale parameter v_0, Smyth [22] showed that the moderated t-statistic and residual sample variance are independent, with the following distributions:

$$\begin{aligned} \hat{\sigma}_g^2 &\sim s_0^2 F_{d_g,d_0}, \\ \tilde{t}_g | z_g = 0 &\sim t_{d_0+d_g}, \\ \tilde{t}_g | z_g = 1 &\sim (1 + v_0/v_g)^{1/2} t_{d_0+d_g}, \end{aligned}$$

where $F(.)$ and $t(.)$ are the central F and t distributions. Based on these results, we can define the emission probabilities as

$$\begin{aligned} f(\tilde{t}_g, \hat{\sigma}_g^2 | z_g = 1) = s_0^2 F_{d_g,d_0} (1 + v_0/v_g)^{1/2} t_{d_0+d_g}, \\ f(\tilde{t}_g, \hat{\sigma}_g^2 | z_g = 0) = s_0^2 F_{d_g,d_0} t_{d_0+d_g}. \end{aligned}$$

The parameters associated with these two density functions are $\Theta = (s_0^2, d_0, v_0)$.

17.4 HMRF Models for Network-Based Analysis of SNP Data

We reviewed in previous section some problems and the HMRF formulations in analysis of microarray gene expression data. We consider in this section the use of HMRF model for large-scale genetic association studies in order to account for linkage disequilibrium (LD) among the SNPs. Suppose we have m cases and n controls that are genotyped over a set of p SNPs. Let $S = \{1, \ldots, p\}$ denote the SNP index. We want to determine which SNPs in S are associated with disease. Let $Y = (Y_1, \ldots, Y_s, \ldots, Y_p)$ be the observed genotype data for the p SNPs, where Y_s itself is a vector $Y_s = (y_{s1}, \ldots, y_{sm}; y_{s(m+1)}, \ldots, y_{s(m+n)})$, where y_{si} is the observed genotype for the ith individual at the sth SNP. In large-scale GWAS studies, many SNPs are in high LD, at least locally.

17.4.1 Weight LD Graphs and MRF Model

The typical single SNP analysis often ignores the LD among these SNPs. Li et al. [16] proposed a HMRF model to take into account the LD information in identifying the disease-associated SNPs. We first construct a weighted undirected LD graph G based on pair-wise LD information derived from the data or from the HapMap project. Specifically, an edge between SNPs s and s' is drawn with weight

$$w_{ss'} = I(r_{ss'}^2 > \tau) r_{ss'}^2,$$

if $w_{ss'} \neq 0$, where $I(.)$ is the indicator function, $r^2_{ss'}$ is the r^2 measurement of LD between SNPs s and s' and τ is a pre-determined cutoff value.

For a given SNP s, we then define a random indicator variable as

$$z_s = \begin{cases} 1 \text{ if SNP } s \text{ is associated with the disease} \\ 0 \text{ if SNP } s \text{ is not associated with the disease.} \end{cases}$$

For two SNPs s and s' that are linked on the LD graph, i.e., if the r^2 between these two SNPs are greater than τ, we expect that z_s and $z_{s'}$ are dependent. As before, joint probability function for $z = (z_1, \ldots, z_p)$ can be specified by a MRF model,

$$p(z; \Phi) \propto \exp(\gamma \sum_{s=1}^{p} z_s + \eta \sum_{s \sim s'} w_{s,s'} I(z_s = z_{s'})),$$

where γ and $\eta \geq 0$ are the two model parameters, and β measures dependencies of z_s for SNPs in LD. In this model, the parameter $\beta > 0$ encourages the SNPs that are in LD to have similar values of z_s. This is in contrast to the hidden Markov model where some time or spatial order of the SNPs has to be assumed. The conditional association state for SNP s, given the states of all neighboring SNPs is

$$p(z_s | z_{N_s}; \Phi) \propto \exp(\gamma z_s + \eta \sum_{s' \in N_s} w_{s,s'} I(z_s = z_{s'})),$$

where N_s represents the neighbors of the SNP s on the LD graph.

17.4.2 An Empirical Bayes Model for Genotype Data

In order to specify the emission probability function $f(Y_s|z_s)$, let $\theta_s = (\theta_{s1}, \theta_{s2}, \theta_{s3})$ be the genotype frequencies at the sth SNP in the case population, and $\rho_s = (\rho_{s1}, \rho_{s2}, \rho_{s3})$ be the genotype frequencies at the sth SNP in the control population, for genotype values of 0, 1 and 2, respectively. We assume that both of these frequencies across all the SNPs have a Dirichlet prior with parameter $\alpha = (\alpha_1, \alpha_2, \alpha_3)$,

$$f(\theta_s) = f(\theta_{s1}, \theta_{s2}, \theta_{s3}) = \frac{\Gamma(\sum_{j=1}^{3} \alpha_j)}{\prod_{j=1}^{3} \Gamma(\alpha_j)} \prod_{j=1}^{3} \theta_{sj}^{\alpha_j - 1}.$$

The same prior is also assumed for ρ_s. For SNP s, let $y_{s+} = (y_{s+,1}, y_{s+,2}, y_{s+,3})$ denote observed genotype counts data in the m cases and $y_{s-} = (y_{s-,1}, y_{s-,2}, y_{s-,3})$ denote the observed genotype counts data in the n controls. So if SNP s is not associated with the disease, cases should have the same genotype frequencies as the controls. The combined genotype counts data $y_{s0} = y_{s+} + y_{s-}$ are generated from a trinomial distribution with the genotype frequencies of $\theta_s = (\theta_{s1}, \theta_{s2}, \theta_{s3})$. Thus, given $z_s = 0$ the probability of the combined genotype count data is

$$f(Y_s|z_s = 0) = \frac{\Gamma(\sum_{j=1}^{3}\alpha_i)\prod_{i=1}^{3}\Gamma(\alpha_i + y_{s+,i} + y_{s-,i})}{\prod_{i=1}^{3}\Gamma(\alpha_i)\Gamma(\sum_{j=1}^{3}(\alpha_i + y_{s+,i} + y_{s-,i}))}.$$

On the other hand, if SNP s is associated with the disease, i.e., when $z_s = 1$, cases and controls should have different genotype frequencies, in which case we have

$$\begin{aligned} f(Y_s|z_s = 1) = & \frac{\Gamma(\sum_{j=1}^{3}\alpha_i)\prod_{i=1}^{3}\Gamma(\alpha_i + y_{s+,i})}{\prod_{i=1}^{3}\Gamma(\alpha_i)\Gamma(\sum_{j=1}^{3}(\alpha_i + y_{s+,i}))} \\ & \times \frac{\Gamma(\sum_{j=1}^{3}\alpha_i)\prod_{i=1}^{3}\Gamma(\alpha_i + y_{s-,i})}{\prod_{i=1}^{3}\Gamma(\alpha_i)\Gamma(\sum_{j=1}^{3}(\alpha_i + y_{s-,i}))}. \end{aligned}$$

The parameters in these two emission probability functions are $\Theta = (\alpha_1, \alpha_2, \alpha_3)$, the hyperparameters in the Dirichlet prior distribution for the genotype frequencies.

17.5 ICM Algorithm, Gibbs Sampling and FDR Controls

For all the models presented in previous sections, the goal is to infer the true state z^* for all p genes or SNPs. One simple approach to this problem is through the use of iterative conditional modes (ICM) algorithm of Besag [2, 3]. However, we need to carry out the parameter estimation simultaneously, including the parameters in the MRF model $\Phi = (\gamma, \eta)$ and the parameters in the emission probabilities Θ. For simple models presented in Sects. 17.3.1 and 17.4, these parameters can be estimated simultaneously during the ICM interactions [16, 30]. For more complex models presented in Sects. 17.3.2 and 17.3.3, a combination of the method of moments and ICM algorithm can be used to estimate the parameters.

1. For parameters in Θ that do not depend on z, denoted here by Θ_1, we use the method of moments of Smyth [22] or Tai and Speed [24] to obtain estimates of these parameters.

2. Obtain an initial estimate $\hat{z}$ of the true states z^* based on the p-values from the standard single-gene tests.

3. Estimate $\Theta_2 = \Theta \backslash \Theta_1$, which maximizes the likelihood

$$l(\mathbf{O}|\hat{z}; \hat{\Theta}_1, \Theta_2) \propto \Pi_{g=1}^{p} f(\mathbf{O}_g|z_g; \hat{\Theta}_1, \Theta_2),$$

where $\mathbf{O}$ represents the data observed and $\mathbf{O}_g$ represents the data observed for the gth gene and $f(O_g|z_g; \hat{\Theta}_1, \Theta_2)$ is the emission probability function defined in previous sections. Note here we make the conditional independence assumption that given z_gs, O_gs are independent.

4. Estimate Φ by the value $\hat{\Phi}$, which maximizes the pseudolikelihood $pl(\hat{z}; \Phi)$ based on the current states $\hat{z}$, where

$$\begin{aligned} pl(z; \Phi) &= \sum_{g=1}^{p} p_g(z_g|z_{\partial_g}; \Phi) \\ &= \sum_{g=1}^{p} \frac{\exp\{\gamma z_g - \eta\mu_g(1 - z_g)/m_g\}}{\exp\{\gamma - \eta\mu_g(0)/m_g\} + \exp\{-\eta\mu_g(1)/m_g\}}. \end{aligned}$$

5. Carry out a single cycle of ICM based on the current $\hat{z}$, $\hat{\Theta}$, and $\hat{\Phi}$ to obtain a new $\hat{z}$. Specifically, for $g = 1, \ldots, p$, update z_g, which maximizes

$$P(z_g|Y, \hat{z}_{\partial_g}) \propto f(\mathbf{O}_g|z_g; \hat{\Theta}) p_g(z_g|\hat{z}_{\partial_g}; \hat{\Phi}), \tag{17.4}$$

subject to $z_g = 1$ or $z_g = 0$.

6. Go to step 3 until approximate convergence of all the parameters. In particular, we stop the iterations when the maximum of the relative changes of the parameter estimates is smaller than a small value ϵ.

After obtaining the parameter estimates $\hat{\Theta}$ and $\hat{\Phi}$ based on the algorithm outlined above, we then carry out a Gibbs sampling procedure to sample $z_g, g = 1, \ldots, p$ given the data using the conditional probability defined in Eq. 17.4 and obtain posterior probabilities of $q_g = Pr(z_g = 0|\mathbf{O}; \hat{\Theta}, \hat{\Phi}), g = 1, \ldots, p$. The resulting posterior probabilities are then used to determine which genes are affected by the phenotype and those relevant genes can be mapped back to the network to identify the subnetworks. In addition, we can estimate the false discovery rate (FDR) based on these posterior probabilities [23]. Specifically, consider p null hypotheses, H_{0g}, let $q_{(1)}, \ldots, q_{(p)}$ be the order values of the posterior probabilities and $H_{(01)}, \ldots, H_{(0p)}$ be the corresponding null hypotheses. The data-driven FDR procedure can be defined as:

$$\text{let } l = \max\left\{ i : \frac{1}{i}\sum_{g=1}^{i} q_{(g)} \leq \alpha \right\}, \text{then we reject all } H_{(0i)}, i = 1, \ldots, l.$$

If all the parameters are known, using the same argument as in Sun and Cai [23], we can show that this procedure can indeed control the FDR at α or smaller under the assumed models. It is however unclear whether this still holds for the data-driven procedure due to fact that the theoretical properties of the parameter estimates are unknown.

17.6 Discussions, Applications and Future Directions

With the increase in availability of human regulatory networks and protein interaction networks, the focus of bioinformatics research has shifted from understanding networks encoded by model species to understanding the pathways and networks underlying human diseases [10]. In order to incorporate the prior biological network information into the analysis of gene expression data related to human diseases, we have reviewed in this paper the network-based empirical Bayes methods for analysis of gene expression and SNP data. Different from the commonly used empirical Bayes methods for analysis of microarray gene expression data that assume independence among the genes, our proposed method imposes dependency among the latent indicator variables using a simple discrete Markov random field model defined on a known regulatory network. In this section, we present a brief summary of applications of these methods to analyses of several human gene expression data using KEGG regulatory network. Finally, we discuss a few other related areas that require new statistical methodological research.

17.6.1 Application to analysis of human microarray gene expression data

We have experienced the application of these methods to analysis of human gene expression data using the KEGG regulatory pathways and have obtained encouraging results. Wei and Li [29] applied the method for analysis of breast cancer gene expression data to identify the connected KEGG subnetworks that are associated with breast cancer recurrence. Wei et al. [31] applied the method to analysis of a time course gene expression data of TrkA- and TrkB-transfected neuroblastoma cell lines and identified genes and subnetworks on MAPK, focal adhesion and prion disease pathways that may explain cell differentiation in TrkA-transfected cell lines. Li et al. [15] applied the methods to analysis of brain ageing data and identified several aging related molecular modules, including subnetwork includes fibroblast growth factors (FGF1, FGF2, FGF12, FGF13) and their receptor (FGFR3) and the mitogen-activated protein kinase (MAPK) (MAPK1 and MAPK9) and the specific MAPK kinase (MAP2K). Li et al. [16] applied the HMRF model with weighted LD graphs to a genetic association study of neuroblastoma.

It should be noted that the proposed methods can be applied to other relevant pathways such as human protein-protein interaction networks. Chuang et al. [5] used the protein-protein interaction network for breast cancer classification and obtained encouraging results in getting replicable molecular modules and better predictions.

Since our current knowledge of the genetic pathways of humans is still very limited, our proposed method depends on the validity of the regulatory networks used. One limitation of the proposed method is that the gene dependency provided by these prior networks may not be reflected at the gene expression levels. If this is the case, we should expect that the estimate of the dependency parameter η in the MRF model to be small, then the network information will not contribute too much and the results should be similar to the standard empirical Bayes analysis. It would be interesting to test the ideas in this paper on other types of biological networks such as protein-protein interaction networks.

17.6.2 Future Directions

The methods presented in this paper model the dependency of the distributions of the gene- or SNP- specific data conditioning on experimental covariates or patient-specific phenotypes, where the gene expression or genotype data are treated as response variables. The goal of such analysis is to identify the covariate- or phenotype-associated genes or SNPs. A related but very different problem is to treat the phenotype as the response and the high-dimensional gene expression data or SNPs as covariates in high-dimensional regression frameworks. This is often done when the sample sizes are very large. The goals of such regression analysis are often two-fold: to identify the genes/SNPs that are predictive to the phenotypes and to build predictive models. One interesting statistical question is how to incorporate the prior network information into such high-dimensional regression analysis. One approach to this problem is in the framework of penalized regression where a penalty function can be defined to account for the expected smoothness of the regression coefficients on the network [14, 33]. Alternatively, a Bayesian variable selection approach can also be developed ([18]). However, much work needs to be done in this important area.

We presented one way of accounting for LD among the SNPs in genome-wide association studies using the idea of weighted LD graph. However, there is still a computational limitation to consider all the SNPs in typical GWAS. One possibility is to perform the analysis chromosome-by-chromosome. An interesting extension for analysis of GWAS data is to incorporate the prior biological network information assuming that whether a gene is associated with disease directly depends on an association status of genes within the same pathway. Noticing that different genes can have different numbers of SNPs, we should first have a method to summarize the SNP variations within a gene into a gene-level statistic. One simple approach is to perform the principal components (PC) analysis and to obtain the first several PCs for each gene. It would be interesting to explore whether such a procedure can indeed lead to identifying more true disease-associated genes.

We have used the maximum pseudo-likelihood estimation (MPLE) during the ICM algorithm to estimate the parameters related to the MRF Φ. An alternative to learn the MRF parameters is to use the iterative proportional fitting (IPF) algorithm.

Comparisons between these two procedures in the settings considered in this paper deserve further studies. Theoretical supports for MPLE and the IPF procedure and the corresponding data-adaptive FDR control procedure are also needed. There is also a need for statistical methods to assess the significance of the subnetworks that are identified by the HMRF models.

Finally, real biological pathways/networks are more complicated than simple undirected graphs. The nodes of such networks may represent different biological quantities, the links may represent different biological interactions and the real networks are often directed. In addition, besides gene expression data, many types of genomic data are being generated. How to integrate these data with detailed prior biological network information raises many challenging statistical problems.

Acknowledgements This research was supported in part by NIH grants ES009911 and CA127334.

References

1. Aitchison, J., & Dunsmore, I. R. (1975). *Statistical prediction analysis*. London: Cambridge University Press.
2. Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*, *36*, 192–225.
3. Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of Royal Statistical Society B*, *48*, 259–302.
4. Bader, G. D., Donaldson, I., Wolting, C., Ouellette, B. F., Pawson, T., & Hogue, C. W. (2001). BIND–The Biomolecular Interaction Network Database. *Nucleic Acids Research*, *29*, 242–245.
5. Chuang, H. Y., Lee, E., Liu, Y. T., Lee, D., & Ideker, T. (2007). Network-based classification of breast cancer metastasis. *Molecular Systems Biology*, *3*, 140.
6. Deng, M., Tu, Z., Sun, F., & Chen, T. (2004). Mapping gene ontology to proteins based on proteinprotein interaction data. *Bioinformatics*, *20*, 895–902.
7. Deng, M., Zhang, K., Mehta, S., Chen, T., & Sun, F. (2003). Prediction of protein function using proteinprotein interaction data. *Journal of Computational Biolology*, *10*, 947–960.
8. Efron, B., Tibshirani, R., Storey, J. D., & Tusher, V. (2001). Empirical Bayes analysis of a microarray experiment. *Journal of the American Statistical Association*, *96*, 1151–1160.
9. Hong, F. X., & Li, H. (2006). Functional hierarchical models for identifying genes with different time-course expression profiles. *Biometrics*, *62*, 534–544.
10. Ideker, T., & Sharan, R. (2008). Protein networks in disease. *Genome Research*, *18*, 644–652.
11. Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., & Speed, T. P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, *4*, 249–264.
12. Kanehisa, M., & Goto, S. (2002). KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, *28*, 27–30.
13. Kendziorski, C.M., M.A. Newton, H. Lan, & M.N. Gould (2003). On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles. *Statistics in Medicine*, *22*, 3899–3914.
14. Li, C., & Li, H. (2008). Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics*, *24*, 1175–1182.
15. Li, C., Wei, Z., & Li, H. (2010). Network-based empirical Bayes methods for linear models with applications to genomic Data. *Journal of Pharmaceutical Statistics*, *20*, 209–222.

16. Li, H, Wei, Z., & Maris, J. (2010). A hidden Markov random field model for genome-wide association studies. *Biostatistics*, *11*, 139–150.
17. Lonnstedt, I., & Speed, T. P. (2002). Replicated microarray data. *Statistica Sinica*, *12*, 31–46.
18. Monni, S., & Li, H. (2010). Bayesian analysis for graph-structured genomics data. In M. Chen, D. K. Dey, P. D. Mueller, & Y. Ye (Eds.), *Frontier of statistical decision making and bayesian analysis – In honor of James O. Berger*.
19. Newton, M.A., C.M. Kendziorski, C.S. Richmond, F.R. Blattner, & K.W. Tsui (2001). On differential variability of expression ratios: Improving statistical inference about gene expression changes from microarray data. *Journal of Computational Biology*, *8*, 37–52.
20. Peri, S., Navarro, J. D., Amanchy, R., Kristiansen, T. Z., Jonnalagadda, C. K., Surendranath, V., Niranjan, V., Muthusamy, B., Gandhi, T. K., Gronborg, M., Ibarrola, N., Deshpande, N., Shanker, K., Shivashankar, H. N., Rashmi, B. P., Ramya, M. A., Zhao, Z., Chandrika, K. N., Padma, N., Harsha, H. C., Yatish, A. J., Kavitha, M. P., Menezes, M., Choudhury, D. R., Suresh, S., Ghosh, N., Saravana, R., Chandran, S., Krishna, S., Joy, M., Anand, S. K., Madavan, V., Joseph, A., Wong, G. W., Schiemann, W. P., Constantinescu, S. N., Huang, L., Khosravi-Far, R., Steen, H., Tewari, M., Ghaffari, S., Blobe, G. C., Dang, C. V., Garcia, J. G., Pevsner, J., Jensen, O. N., Roepstorff, P., Deshpande, K. S., Chinnaiyan, A. M., Hamosh, A., Chakravarti, A., & Pandey, A. (2003). Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Research*, *13*, 2363–2371.
21. Sharan, R., Ulitsky, I., & Shamir, R. (2007). Network-based prediction of protein function. *Molecular Systems Biology*, *3*(88).
22. Smyth, G. K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, *3*(1), Article 3.
23. Sun, W., & Cai, T. (2009). Large-scale multiple testing under dependency. *Journal of the Royal Statistical Society, Series B*, *71*, 393–424.
24. Tai, Y. C., & Speed, T. (2006). A multivariate empirical Bayes statistic for replicated microarray time course data. *Annals of Statistics*, *34*, 2387–2412.
25. Ulitsky I., Karp, R. M., & Shamir, R. (2008). Detecting disease-specific dysregulated pathways via analysis of clinical expression profiles. In *Proceeding of RECOMB 2008* (pp. 347–359). Berlin: Springer.
26. Ulitsky, I., & Shamir, R. (2008). Detecting pathways transcriptionally correlated with clinical parameters. *Proceedings of the 7th annual international conference on computational systems bioinformatics (CSB 08)* (pp. 249–258). London, UK: Imperial College Press.
27. Ulitsky, I., & Shamir, R. (2009). Identifying functional modules using expression profiles and confidence-scored protein interactions. *Bioinformatics*, *25*, 1158–1164.
28. Wei, P., & Pan, W. (2008). Incorporating gene networks into statistical tests for genomic data via a spatially correlated mixture model. *Bioinformatics*, *24*, 404–411.
29. Wei, Z., & Li, H. (2007). A Markov random field model for network-based analysis of genomic data. *Bioinformatics*, *23*, 1537–1544.
30. Wei, Z., & Li, H. (2008). A hidden spatial-temporal Markov random field model for network-based analysis of time course gene expression data. *Annals of Applied Statistics*, *2*(1), 408–429.
31. Wei, Z., Minturn, J. E., Rappaport, E., Brodeur, G., and Li, H. (2008). Incorporation of genetic pathway information into analysis of multivariate gene expression data. In A. Yakovle, L. Klebanov, & G. Gaile (Eds.), *Statistical methods for microarray data analysis*. Unpublished manuscript.
32. Yuan, M., & Kendziorski, C. (2006). Hidden Markov models for microarray time course data under multiple biological conditions (with discussion). *Journal of the American Statistical Association*, *101*(476), 1323–1340.
33. Zhu, Y., Pan, W., & Shen, X. (2009). Support vector machines with disease-centric network penalty for high dimensional microarray data. *Statistics and its Inference*, *2*(3), 257–269.

Chapter 18
Review of Weighted Gene Coexpression Network Analysis

Tova Fuller, Peter Langfelder, Angela Presson, and Steve Horvath

Abstract We survey key concepts of weighted gene coexpression network analysis (WGCNA), also known as weighted correlation network analysis, and related data analysis strategies. We describe the construction of a weighted gene coexpression network from gene expression data, identification of network modules and integration of external data such as gene ontology information and clinical phenotype data. We review Differential Weighted Gene Coexpression Network Analysis (DWGCNA), a method for comparing and contrasting networks constructed from qualitatively different groups of samples. DWGCNA provides a means for measuring not only differential expression but also differential connectivity. Further, we show how to incorporate genetic marker data with expression data via Integrated Weighted Gene Coexpression Network Analysis (IWGCNA). Lastly, we describe R software implementing WGCNA methods.

18.1 Introduction

The merging of network theory with gene expression data analysis techniques has spawned a new field: gene coexpression network analysis. Genes with similar expression patterns may participate in pathways and in regulatory and signaling circuits [13], and their products may form complexes. Constructing a network of genes based on coexpression facilitates the understanding of such phenomena and identification of their key players.

Gene coexpression networks are also referred to as 'association', 'correlation' or 'influence' networks. They have been used to describe the transcriptome in many organisms, for example, in yeast, flies, worms, plants, mice and humans [4, 5, 7, 19, 46, 47, 49, 50, 54, 55]. Network methods have also been used for 'standard' microarray data analysis tasks such as gene filtering [16, 23, 35, 54], sample classification and outcome prediction [9, 43].

T. Fuller, P. Langfelder, A. Presson, and S. Horvath (✉)
Human Genetics and Biostatistics, University of California, Los Angeles
e-mail: shorvath@mednet.ucla.edu

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_18,
© Springer-Verlag Berlin Heidelberg 2011

Here we will describe weighted gene coexpression network analysis (WGCNA) [22, 23, 30, 54], a systems biology method for describing the correlation patterns among genes across microarray samples. WGCNA can be used for finding clusters (modules) of highly correlated genes, for summarizing such clusters using the module eigengene or an intramodular hub gene, for relating modules to one another and to external sample traits (using eigengene network methodology) and for calculating module membership measures. Network based gene screening methods can be used to identify candidate biomarkers or therapeutic targets. These methods have been successfully applied in various biological contexts such as cancer, mouse genetics and yeast genetics.

Before describing two specific data analysis strategies (the differential network analysis DWGCNA and the Marker Integrated WGCNA), we briefly review the key concepts of the WGCNA framework.

18.1.1 Constructing a Weighted Coexpression Network

WGCNA uses network terminology to describe coexpression, or correlation patterns among probe set or ‘gene’ expression profiles. For the purposes of this chapter, we do not distinguish between probe sets and genes. The nodes of a gene coexpression network correspond to genes, labeled by indices $i, j = 1, 2, \ldots, n$, and each edge is determined by the pairwise correlation between two gene expression profiles. The network can be specified by its *adjacency matrix* **A**, a symmetric matrix with entries a_{ij} in [0, 1] that encode the strength of the link between genes i and j. It is useful to define the adjacency **A** in terms of *coexpression similarity* $s_{ij} = |\mathrm{cor}(x_i, x_j)|$. This defines an unsigned network in which positive and negative correlations are treated equally. Optionally, one may also want to preserve the sign of the correlation, using a *signed* similarity defined as $s_{ij} = (1 + \mathrm{cor}(x_i, x_j))/2$. Signed and unsigned similarities differ in how they treat negatively correlated genes: genes with a high negative correlation (close to -1) will have a low similarity in a signed network but a high similarity in an unsigned network.

In an unweighted network, the adjacency can be defined by hard thresholding the coexpression similarity s_{ij}: genes i and j are linked ($a_{ij} = 1$) if the absolute correlation between their expression profiles exceeds a pre-defined constant τ called the *hard threshold*. While unweighted networks are widely used, they do not reflect the continuous nature of the underlying coexpression information and may thus lead to an information loss. In contrast, weighted networks reflect the continuous nature of coexpression by allowing the adjacency to take on continuous values between 0 and 1. A weighted network adjacency can be defined by raising the coexpression similarity s_{ij} to a power $\beta \geq 1$, referred to as the ‘soft threshold’. By raising the similarity to a power, the weighted gene coexpression network construction emphasizes high correlations at the expense of low correlations. In summary, a weighted unsigned network is defined by

$$a_{ij} = s_{ij}^{\beta}.$$

18.1.2 Gene Significance

To incorporate external information into the coexpression network, we make use of gene significance (GS) measures. Abstractly speaking, the higher the absolute value of GS_i, the more biologically significant is the i-th gene with regard to a given application. GS_i could encode pathway membership (for example, it equals 1 if the gene is a known apoptosis gene and 0 otherwise), or it could encode knockout essentiality.

When a sample trait is available (for example, body weight), we often use the absolute value of the correlation coefficient between the trait and the gene expression profiles to define a trait based gene significance measure $GS_i^{trait} = |\mathrm{cor}(x_i, trait)|$. It is straightforward to calculate the corresponding p-value using a correlation test or a regression model. Alternatively, if a p-value has been defined for each gene, one can define a gene significance measure as the negative log of the p-value, $GS_i = -\log(\text{p-value}(i))$. The gene significance can take on positive or negative values, with $GS_i = 0$ indicating that the gene is not significant with regard to the biological question of interest.

18.1.3 Network Modules

A major step in our analysis is to cluster genes into network modules based on their coexpression. Most standard clustering methods require a distance, or dissimilarity, measure, where highly coexpressed genes have a small dissimilarity. For example, one could use the adjacency-based dissimilarity measure $dissAdj_{ij} = 1 - a_{ij}$. If larger and more robust modules are desired, one can use a dissimilarity measure based on the topological overlap matrix (TOM) [39, 54]:

$$dissTOM_{ij} = 1 - TOM_{ij} = 1 - \frac{\sum_{u \neq i} a_{iu} a_{uj} + a_{ij}}{\min(k_i, k_j) + 1 - a_{ij}},$$

where $k_i = \sum_{u \neq i} a_{ui}$ denotes the network connectivity. TOM combines the connection strength between a pair of genes with their connections to other 'third party' genes, and has been shown to be a highly robust measure of network interconnectedness (proximity) [32, 53].

The dissimilarity measure of choice is used as input in average linkage hierarchical clustering. Modules are then defined as branches of the resulting cluster tree. Toward this end, a flexible ('dynamic') branch cutting method has been implemented [31]. This module detection procedure has been found useful in many applications [6, 12, 16, 19, 23, 28, 35, 36, 38, 51].

It is often useful to summarize the expression profiles of all genes in a module using a single representative expression profile. For this purpose, we define the *module eigengene E* as the first principal component of the standardized expression profiles of a given module [22, 29]. The eigengene can be considered a weighted average of the module gene expressions. When a sample trait y is available, one can correlate the module eigengenes with this trait. The correlation coefficient or its corresponding p-value is referred to as the *eigengene significance*.

For each module we also define its *module significance* as the average absolute gene significance for all genes in the module. When gene significance is defined as the correlation of gene expression profiles with an external trait y, module significance tends to be highly related to the corresponding eigengene significance.

By relating only a few modules to the trait data rather than thousands of genes, WGCNA alleviates the multiple testing problem inherent in gene expression data analysis. Because the modules may correspond to biological pathways, focusing the analysis on modules amounts to a biologically motivated data reduction scheme.

18.1.4 Network Concepts and Connectivity

The term *network concepts* refers to functions of the adjacency matrix and/or a gene significance measure that characterize topological properties of the gene network, both globally and at the level of individual genes. For example, the average adjacency is referred to as network density and the average gene significance across genes is referred to as network significance [22].

An important network concept is the connectivity k_i (also known as degree) that measures how connected the i-th gene is with other genes in the network. The *whole network connectivity*, often simply called *connectivity*, is defined as the sum of connection strengths (adjacencies) of gene i with all other network genes:

$$k_i = \sum_{u \neq i} a_{ui} .$$

By definition, genes inside coexpression modules tend to be highly connected – in other words, they tend to have high whole network connectivity.

Network connectivity depends on many biological and technical factors. As a result, different networks will inherently have different connectivities. To facilitate comparison of connectivity values between two different networks, we define the *scaled whole network connectivity* as

$$K_i = \frac{k_i}{\max_i(k_i)}, \tag{18.1}$$

which is the connectivity divided by the maximum connectivity in the network.

The *intramodular connectivity* $k_{IM,i}$ measures how connected, or coexpressed, the i-th gene is with respect to the genes of a particular module. Intramodular connectivity is calculated as the sum of the adjacencies within the module of interest:

$$k_{IM,i} = \sum_{u \in Module} a_{ui}.$$

If an eigengene is available (for example, E^q denotes the eigengene of module q), one can define the eigengene based measure of intramodular connectivity (also known as fuzzy module membership measure) as

$$k^q_{ME,i} = MM^q(i) = \text{cor}(x_i, E^q), \tag{18.2}$$

which measures how correlated gene i is with the eigengene of module q. The module membership measure takes on values in $[-1, 1]$. If $MM^q(i)$ is close to 0, the i-th gene is not part of module q. On the other hand, if $MM^q(i)$ is close to 1 or -1, it is highly connected to genes in module q. The module membership can be defined for all genes (irrespective of whether they were used in network construction or not). Module membership has been used to annotate genes with respect to cell type specific modules in the human brain transcriptome [36].

While there is a close relationship between the eigengene based connectivity $k_{ME,i}$ and intramodular connectivity $k_{IM,i}$ for a given module, we prefer $k_{ME,i}$ for the following reasons: (1) it is naturally scaled to take on values between -1 and 1, (2) one can use a correlation test to calculate a corresponding p-value for a gene's module membership, (3) it can be used in signed networks to identify genes that are anti-correlated with a given module eigengene and (4) k_{ME} can be computed for any gene on the array (not just genes used in the network construction). In practice, we find that intramodular and module eigengene based connectivity are highly correlated [22].

18.1.5 Are Hub Genes Important?

Hub genes are highly connected genes, or put another way, genes that interact with many other genes. The precise definition of hub gene status depends on the connectivity measure and a threshold for this measure. For example, the top 20% most highly connected genes could be referred to as hub genes, but other thresholds have also been used. Network theorists have long studied the relationship between the biological significance of a gene and its centrality or network connectivity. Clearly, the precise definition of biological significance (i.e., the choice of the gene significance measure) depends on the research question and the application. Several network articles have pointed out that highly connected hub nodes are often central to network architecture [1, 2, 7, 20]. While theses genes are not always critical in higher organisms, knock-out experiments in yeast have shown that hub genes are essential for survival [6,25]. A theoretical and empirical analysis of hub gene significance with regard to different connectivity measures and different gene significance measures can be found in [22]. That work also provides a geometric characterization of networks in which hub genes are important. These theoretical findings show that intramodular connectivity (as opposed to whole network connectivity) with respect to biologically significant modules can be an important complementary gene screening variable. Intramodular hub genes are centrally located in the module and thus summarize the pathway state. Several publications have demonstrated the importance of intramodular hub genes, for example, in brain cancer [23] or inflammatory response [17].

Several studies have shown that differences in connectivity can be used to identify biologically important genes in coexpression networks of higher organisms [16, 19, 35, 48]. Below, we review a systems-genetic gene screening strategy that combines intramodular connectivity with causality testing scores [38]. This concludes our introduction on the key concepts of WGCNA.

18.2 WGCNA Applications

In this section we will review two WGCNA approaches to complex disease analysis: (1) Differential WGCNA or DWGCNA, which allows one to view systematic differences between different subgroups or species, and (2) Integrated WGCNA, or IWGCNA, which integrates genetic marker information to characterize network relationships as causal or reactive. Figure 18.1 provides an overview of WGCNA, DWGCNA and IWGCNA.

18.2.1 DWGCNA

We now describe differential weighted gene coexpression network analysis, or DWGCNA, which may be useful for identifying gene pathways distinguishing phenotypically distinct groups of samples. Differential network analysis is concerned with identifying both differentially connected and differentially expressed genes. We illustrate this analysis approach using data from a previously studied F_2 intercross between inbred strains C3H/HeJ and C57BL/6J [16, 19]. The liver tissues of 135 female mice were analyzed. We identified the 30 mice at both extremes of the weight spectrum and constructed the first network using the 30 leanest mice and the second network using the 30 heaviest mice. To measure differential gene expression between the lean and the obese mice, we use the absolute value of the Student t-test statistic. To measure differential connectivity, we use

$$\mathit{DiffK(i)} = K^1(i) - K^2(i),$$

where $K^1(i)$ and $K^2(i)$ measure the scaled whole network connectivity (Eq. 18.1) in lean mice (group 1) and obese mice (group 2), respectively.

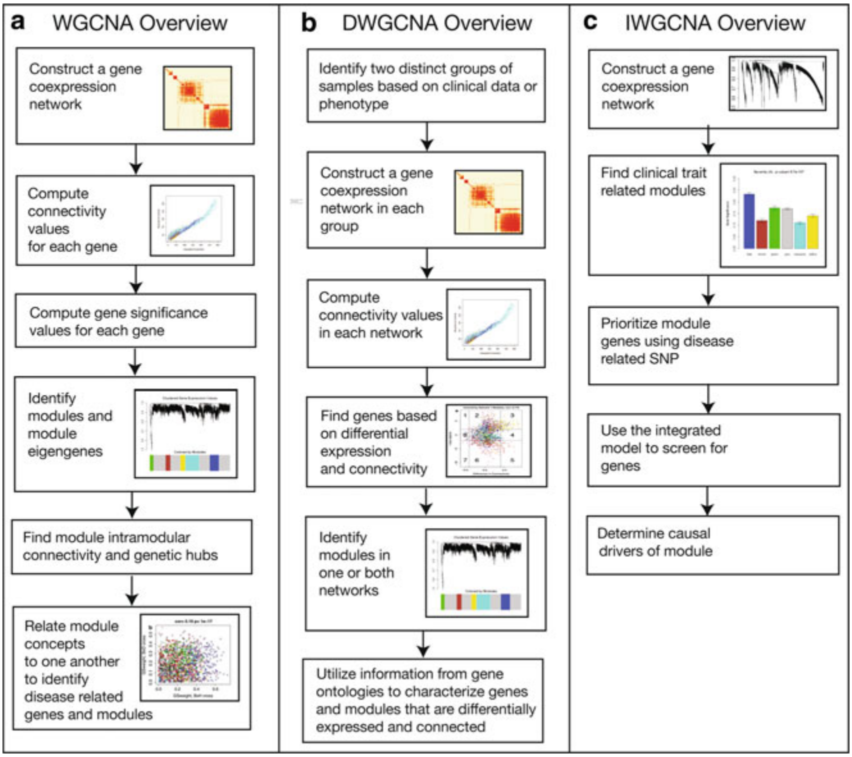

a WGCNA Overview
Construct a gene coexpression network
Compute connectivity values for each gene
Compute gene significance values for each gene
Identify modules and module eigengenes
Find module intramodular connectivity and genetic hubs
Relate module concepts to one another to identify disease related genes and modules
b DWGCNA Overview
Identify two distinct groups of samples based on clinical data or phenotype
Construct a gene coexpression network in each group
Compute connectivity values in each network
Find genes based on differential expression and connectivity
Identify modules in one or both networks
Utilize information from gene ontologies to characterize genes and modules that are differentially expressed and connected
c IWGCNA Overview
Construct a gene coexpression network
Find clinical trait related modules
Prioritize module genes using disease related SNP
Use the integrated model to screen for genes
Determine causal drivers of module

Fig. 18.1 Overview of network methodology. (**a**) Overview of WGCNA. (**b**) Overview of DWGCNA. (**c**) Overview of IWGCNA

18.2.1.1 Identifying Gene Sectors Based on Differential Expression and Connectivity

We hypothesized that changes in connectivity may correspond to large-scale rewiring of the gene coexpresson network in response to environmental changes, physiologic perturbations or genetic variations. Plotting the difference *DiffK* in connectivity between lean and obese mice versus the t-test statistic for differential expression of each gene gives a visual demonstration of how difference in connectivity relates to a t-statistic describing difference in expression between the two networks.

Figure 18.2a shows a scatterplot of *DiffK* versus the Student's t-statistic of differential expression. Eight sectors of the plot with high absolute values of *DiffK* (>0.4) and/or t-statistics (>1.96) are shown. Horizontal lines depict sector boundaries based on t-statistic values, and vertical lines depict boundaries based on *DiffK*. These eight sectors are marked by numbers in Fig. 18.2a. We use a permutation test to determine sector significance. The permutation test contrasts networks built by randomly partitioning the 60 mice into two groups. Figure 18.2b demonstrates

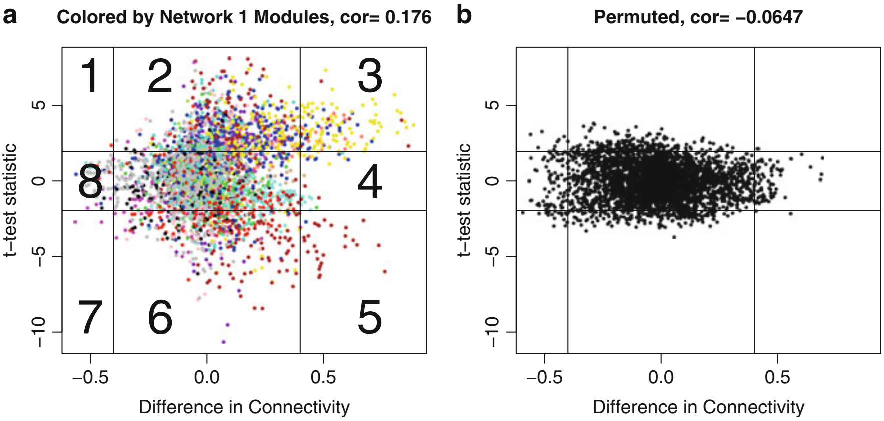


Fig. 18.2 Sector plots of differential network analysis. In (**a**) and (**b**), difference in connectivity (*DiffK*) is plotted on the x-axis, and t-test statistic values are plotted on the y-axis. Horizontal lines indicate a difference in connectivity of −0.4 and 0.4, whereas vertical lines depict a t-statistic value of −1.96 or 1.96. (**a**) Observed *DiffK* and t-statistic values. Genes are colored based on network 1 module definitions. Numbers indicate sectors 1–8. (**b**) Corresponding sector plot for a permuted network where array samples in data sets 1 and 2 were randomly permuted

the the relationship between *DiffK* and t-statistic when network membership is permuted. Based on 1,000 random permutations, sectors 2, 3 and 6 were significant ($p \leq 1.0 \times 10^{-3}$). Membership in sector 5 was significant with $p \leq 1.0 \times 10^{-2}$. Clearly, we find that genes that are differentially connected may or may not be differentially expressed.

18.2.1.2 Functional Enrichment Analysis of Sector 3 and Sector 5 Genes

We used the DAVID database to determine the functional enrichment of 61 sector 3 genes that were both highly connected in network 1 and weakly connected in network 2 [11]. We focused on sector 3 for two reasons. First, sector 3 members had extreme values of *DiffK* as well as high t-statistic values. Also, as one can see from Fig. 18.2a, a high proportion of yellow module genes were found in this sector based on network 1 module definitions. These yellow module genes were weakly connected in network 2, and therefore were annotated as grey module (background) members in a module assignment scheme based on network 2. This result suggests that in a pathophysiologic state (mouse obesity), the yellow module can no longer be found. Genes in the yellow module were enriched for the extracellular region, extracellular space, signaling, cell adhesion and glycoproteins at the $p < 0.05$ level. Furthermore, 12 terms for epidermal growth factor or its related proteins were recovered in the functional analysis.

Sector 5 is analogous to sector 3 in that it contains genes with both extreme differences in connectivity and extreme t-statistic values. After Bonferroni correction, sector 5 genes were enriched for enzyme inhibitor activity, endopeptidase activity, dephosphorylation, protein amino acid dephosphorylation and serine-type endopeptidase inhibitor activity at the $p < 0.05$ level. Two genes were found in all mentioned categories: *Itih1* and *Itih3*.

In summary, DWGCNA identified genes and pathways that are not only differentially expressed, but also differentially connected. This additional information describes the differential wiring of genes. R code and tutorials to reproduce these results are available from http://www.genetics.ucla.edu/labs/horvath/\CoexpressionNetwork/DifferentialNetworkAnalysis/.

18.2.2 IWGCNA

The availability of genetic marker data enables causality testing to identify the genetic drivers underlying the modules and clinical traits of interest. The concept of conducting a causality analysis based on genetic marker data has been explored by several authors [3, 8, 10, 26, 33, 37, 41, 45]. We refer to a weighted gene coexpression network analysis that uses genetic markers in causality testing as 'Marker Integrated WGCNA' or simply as IWGCNA. We review the IWGCNA approach for integrating a weighted gene coexpression network with SNP data to identify a disease-related module and to develop a systems genetic screening strategy that generates testable hypotheses. Furthermore, we use the Network Edge Orienting (NEO) software [3] to show that this screening strategy prefers genes that are causal for the modules.

IWGCNA uses correlation to relate gene expression profiles, genetic markers and clinical traits. Using correlation provides a unified approach for relating variables from disparate data sets. IWGCNA aims to find genes that are (1) significantly related to the clinical trait, (2) highly connected hub genes in a disease related coexpression module and (3) significantly associated with a disease related marker. Figure 18.1c outlines the main steps of IWGCNA.

18.2.2.1 Steps of the IWGCNA Analysis

In this section we review the application of IWGCNA on a chronic fatigue syndrome (CFS) data set and show that it identifies candidate genes whose functions are consistent with results from other CFS studies [38]. We analyzed the phenotype, genotype and expression data from a 4 year longitudinal study conducted by the Centers for Disease Control (CDC) [40]. We focused on 127 patients that were diagnosed with some level of fatigue according to the Intake diagnosis (i.e., we removed the controls). The goal was to find genes and pathways that relate to CFS severity, which is an ordinal outcome with levels mild, moderate and severe.

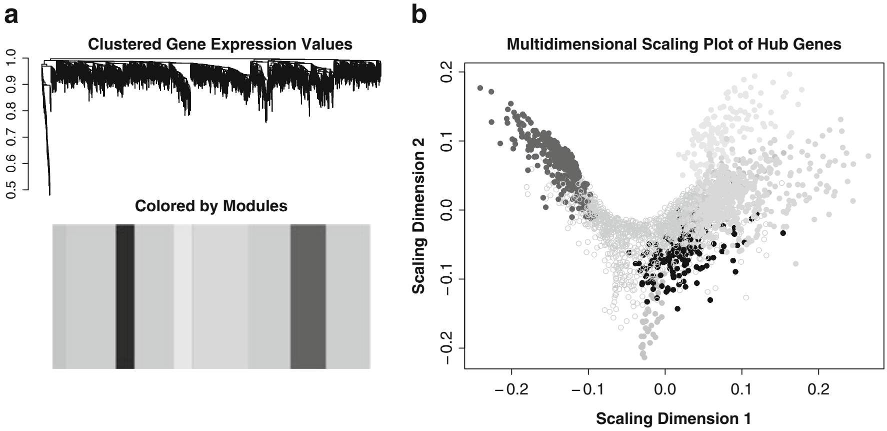


Fig. 18.3 Visualizing the network. (**a**) Hierarchical clustering of the 2,677 most varying and connected genes in a chronic fatigue syndrome data set resulted in five modules. (**b**) A multi-dimensional scaling plot of these genes indicates that the blue module is the most distinct

First, we constructed a coexpression network from the microarray data. Figure 18.3 shows the five modules of coexpressed genes that were identified by WGCNA and a classical multi-dimensional scaling plot of their relative positions. We defined $GS_{severity}(i)$ as the absolute value of the correlation between the CFS severity phenotype and the i-th gene expression x_i: $GS_{severity}(i) = |\text{cor}(x_i, \text{severity})|$. Note that gene significance raised to a power β can be interpreted as the connection strength between severity and the i-th gene expression in a weighted network.

Second, we identified a module related to CFS severity. To arrive at a measure of module significance, we averaged the $GS_{severity}$ values of all genes within a module. The blue module with 299 genes had the highest module significance (average $GS_{severity} = 0.234$, $p = 0.007$) and was selected for further analysis.

Third, we used a severity-related SNP marker to prioritize genes within the blue module. This step required a SNP marker that is associated with both the trait and the trait related module. The genetic marker data consisted of 36 autosomal SNPs located near or within a set of eight genes that were considered biologically relevant for CFS [44]. We chose to focus on SNP rs10784941 located within the *TPH2* tryptophan hydroxylase two gene because it had previously been shown to be associated with chronic fatigue and CFS severity in our data set. To measure the association between a SNP and the gene expression profiles, we defined a SNP-based gene significance measure: $GS_{SNP}(i) = |\text{cor}(x_i, SNP)|$. GS_{SNP} is similar to a single point LOD score as it measures the extent to which a gene is associated with the SNP.

Fourth, we applied an integrated gene screening strategy to identify candidate genes. While a standard gene screening approach would draft a final list of candidate genes based solely on the association between gene expression and the clinical trait ($GS_{severity}$), our integrated strategy additionally used GS_{SNP} and k_{ME}. This approach allowed us to select disease related genes that were implicated by the genetic marker and network connectivity information. Since the blue module was associated with CFS severity, we used it for a module based screening analysis. We selected candidate genes that met the following criteria: (1) moderate association with the $TPH2$ SNP, (2) k_{ME}^{blue} in the top 80% to select intramodular hub genes, (3) $GS_{severity}$ and GS_{TPH2} signs that were consistent in both sexes and (4) moderate association with the severity trait. This strategy resulted in 20 candidate genes.

Fifth, we used causality testing implemented in the Network Edge Orienting (NEO) software [3] to orient network edges. We calculated the $LEO.NB.SingleMarker$ score, which is a relative fitting index that compares the model fitting p-value of the causal model for a gene x_i causing eigengene E to that of the next best competing model. For the edge orientation of $x_i \rightarrow E$, the $LEO.NB.SingleMarker$ is given by

$$LEO(x_i \rightarrow E|SNP) = \log_{10}\left(\frac{p(\text{model } 1 : SNP \rightarrow x_i \rightarrow E)}{\max\begin{pmatrix} p(\text{model } 2 : SNP \rightarrow E \rightarrow x_i), \\ p(\text{model } 3 : x_i \leftarrow SNP \rightarrow E), \\ p(\text{model } 4 : SNP \rightarrow x_i \leftarrow E), \\ p(\text{model } 5 : SNP \rightarrow E \leftarrow x_i) \end{pmatrix}}\right),$$

where the competing models have the following interpretations: model 2 implies that E causes x_i, model 3 implies that the SNP directly affects both x_i and E so that given the SNP they are independent of each other (confounded model), model 4 implies that the SNP and E both affect x_i and model 5 implies that the SNP and x_i both affect E. Genes with a causal relationship to their parent module are highly related to many other genes within the module and are upstream of the module gene expressions. There were 66 causal genes out of the 299 blue module genes, and all but three of our 20 candidate genes were causal for the blue module. A NEO analysis of the combined male and homogenized female samples found that 18 candidate genes were causal. The enrichment for causal genes within our candidate gene set shows that our gene screening strategy favors causal drivers.

The analysis described in steps 1–5 is a biologically motivated gene screening strategy. Gene ontology software showed that the majority of our candidate genes interacted in a cell death pathway.

In summary, we have shown that WGCNA can be combined with genetic marker data to identify disease-related genes, pathways and their causal drivers. IWGCNA of a CFS data set identified candidate genes that interact in a biologically relevant pathway. Integrating gene coexpression networks with allelic association studies may be useful for identifying complex disease genes.

18.3 Software for WGCNA

The WGCNA R software package [30] provides a comprehensive collection of R functions for performing various aspects of weighted correlation network analysis. The package includes functions for the following tasks: (1) network construction, (2) module detection, (3) module and gene selection (screening), (4) calculations of network topological properties, (5) data simulation, (6) visualization and (7) interfacing with external software packages. Along with the R package, we also provide user-friendly R software tutorials. While the methods development was motivated by gene expression data, the underlying data mining approach can be applied to a variety of different settings. The R package, along with its source code and additional material, are freely available at http://www.genetics.ucla.edu/labs/horvath/\CoexpressionNetwork/Rpackages/WGCNA. Here we briefly outline the main functionality of the package.

18.3.1 Category 1: Functions for Network Construction

The WGCNA package provides a host of pairwise coexpression similarity measures for constructing networks including more robust measures of correlation (the biweight midcorrelation [52] or the Spearman correlation). Using a thresholding procedure, the coexpression similarity is transformed into the adjacency. Both hard-thresholding (function `signumAdjacencyFunction`) and soft-thresholding (function `adjacency`) are available. Adjacency functions for both weighted and unweighted networks require the user to choose threshold parameters – for example, by applying the approximate scale-free topology criterion [54]. The package provides functions `pickSoftThreshold` and `pickHardThreshold` that assist in choosing the parameters, as well as the function `scaleFreePlot` for evaluating whether the network exhibits an approximate scale-free topology.

18.3.2 Category 2: Functions for Module Identification

WGCNA identifies gene modules using unsupervised clustering, i.e., without the use of a priori defined gene sets. The user has a choice of several module identification methods. The default method is hierarchical clustering using the standard R function `hclust` [27]. Branches of the hierarchical cluster tree correspond to modules and can be identified using one of a number of available branch cutting methods, such as the constant-height cut or two dynamic branch cut methods [31]. Compared to the static constant-height cut, the height and shape parameters of the dynamic tree cut methods offer improved flexibility for branch cutting and module identification.

One drawback of hierarchical clustering is that it can be difficult to determine how many (if any) clusters are present in the data set. While the default parameters of the dynamic tree cut functions have worked well in several applications, in practice we recommend carrying out a cluster stability/robustness analysis. A coexpression module may reflect a true biological signal (e.g., a pathway) or it may reflect noise (e.g., technical artifacts, tissue contamination or false positives). To test whether the identified modules are biologically meaningful, gene ontology information (functional enrichment analysis) can be used. Toward this end, we provide an R tutorial that describes how to interface WGCNA with relevant software packages and data bases.

18.3.2.1 Summarizing the Expression Profiles of a Module

Several options have been implemented for summarizing the gene expression profiles of a given module. For example, the function `moduleEigengenes` represents the module expressions of the q-th module by the module eigengene E^q. Alternatively, the user can use intramodular connectivity to find the most highly connected intramodular hub gene, which represents the module.

18.3.2.2 Fuzzy Measure of Module Membership

Hierarchical clustering and most other standard clustering methods such as Partitioning Around Medoids (PAM) [27] result in a binary module assignment, i.e., a node is either inside or outside of a module. As we discussed previously, in some applications it may be advantageous to define a continuous, fuzzy measure of module membership for all nodes. Such a measure is particularly useful to identify nodes that lie near the boundary of a module, or nodes that are intermediate between two or more modules. Since we define the fuzzy module membership MM as the eigengene-based connectivity (see Eq. 18.2) we named the corresponding R function `signedKME`.

18.3.2.3 Automatic Blockwise Module Detection

Many microarray experiments report expression levels of tens of thousands of distinct genes (or probes). Building and analyzing a full network among such a large number of nodes can be computationally challenging because of memory size and processor speed limitations. The WGCNA package contains several improvements that address this challenge. The function `blockwiseModules` first pre-clusters nodes into large clusters, referred to as blocks, using a variant of k-means clustering (`projectiveKMeans`). Next, hierarchical clustering is applied to each block and modules are defined as branches of the resulting cluster tree. To synthesize the module detection results across blocks, an automatic module merging step (`mergeCloseModules`) is performed that merges modules whose eigengenes are highly correlated. The time and memory savings of the blockwise approach are often substantial.

18.3.2.4 Consensus Module Detection

When dealing with multiple adjacency matrices representing different networks, it can be interesting to find *consensus modules*, defined as modules that are present in all or most networks [29]. Intuitively, two nodes should be connected in a consensus network only if all input networks agree on that connection. This naturally suggests to define the consensus network similarity between two nodes as the minimum of the input network similarities. For multiple input data sets, it can be useful to replace the minimum by a suitable quantile (e.g., the first quartile). Consensus module detection can be performed step-by-step for maximum control and flexibility, or in one step using the function `blockwiseConsensusModule`. This function calculates consensus modules across given data sets in a blockwise manner analogous to the blockwise module detection in a single data set.

18.3.3 Category 3: Functions for Module and Gene Selection

Finding biologically or clinically significant modules and genes is a major goal of many coexpression analyses. The definition of biological or clinical utility depends on the research question under consideration. We have previously discussed the gene significance (GS) measure, which may be used in gene selection. Similarly, the module significance measure may be used in module selection. Furthermore, genes with high module membership in modules related to traits are natural candidates for further validation studies. This strategy is implemented in the `networkScreening` function.

18.3.4 Category 4: Functions for Studying Topological Properties

Many topological properties of networks can be succinctly described using network concepts, also known as network statistics or indices [12, 22]. Network concepts include whole network connectivity (degree), intramodular connectivity, topological overlap, the clustering coefficient and other network measures described in [12]. The WGCNA package implements several functions, such as `softConnectivity`, `intramodularConnectivity`, `TOMSimilarity`, `clusterCoef` and `networkConcepts`, for computing these measures. Basic R functions can be used to create summary statistics and test their differences across networks. Differential analysis of network measures such as intramodular connectivity may reveal regulatory changes in gene expressions as previously described in the section on DWGCNA [16, 35].

18.3.5 Category 5: Functions for Simulating Microarray Data with Modular Structure

Simple yet sufficiently realistic simulated data is often important for evaluation of novel data mining methods. The WGCNA package includes simulation functions `simulateDatExpr`, `simulateMultiExpr` and `simulateDatExpr5Modules` that result in expression data sets with a customizable modular (cluster) structure. The user can choose the modular structure by specifying a set of seed eigengenes, one for each module, around which each module is built. Module genes are simulated to exhibit progressively lower correlations with the seed which leads to genes with progressively lower intramodular connectivity. The user can specify module sizes and the number of background genes, i.e., genes outside of the modules. The seed eigengenes can be simulated to reflect dependence relationships between the modules (`simulateEigengeneNetwork`).

18.3.6 Category 6: Visualization Functions

Module structure and network connections in the expression data can be visualized in several different ways. For example, the coexpression module structure can be visualized by heatmap plots of gene-gene connectivity that can be produced using the function `TOMplot`. An alternative is a multi-dimensional scaling plot; relationships among modules can be summarized by a hierarchical cluster tree of their eigengenes, or by a heatmap plot of the corresponding eigengene network (`labeledHeatmap`). The package includes several additional functions designed to aid the user in visualizing input data and results.

18.3.7 Category 7: Functions for Interfacing with other Software Packages

We have created R functions and tutorials to integrate WGCNA with other network visualization packages and functional enrichment software. For example, the functions `exportNetworkToVisANT` and `exportNetworkToCytoscape` allow the user to export networks in a format suitable for VisANT [24] and Cytoscape [42], respectively.

Our online R tutorials also show how to interface WGCNA results with gene ontology packages available directly in R, e.g., GOSim [15]. Many gene ontology based functional enrichment analysis packages simply take lists of gene identifiers as input.

18.3.8 Tutorials

We provide a comprehensive set of online tutorials that guide the user through the major steps of correlation network analysis. The tutorials provide R code the user can copy and paste into an R session, along with comments and explanations of both the input and output. The tutorials cover the following major topics: correlation network construction, step-by-step and automatic module identification, consensus module detection, eigengene network analysis, differential network analysis, interfacing with external software packages and data simulation. The tutorials use both simulated and real gene expression data sets.

18.4 Discussion

Weighted gene coexpression network analysis (WGCNA) complements other network analysis approaches such as gene network enrichment analysis [34] and functional analysis of gene coexpression networks [21]. While most other approaches focus on unweighted networks, WGCNA preserves the continuous coexpression information. That being said, the WGCNA R package implements methods for both weighted and unweighted correlation networks.

While WGCNA is a powerful microarray analysis tool, users should also be aware of its limitations. First, WGCNA assumes that the microarray data have been properly pre-processed and normalized. To normalize the expression data, several R functions have been implemented in Bioconductor packages [18]. Although all normalization methods are mathematically compatible with WGCNA, we recommend using the biologically most meaningful normalization method with respect to the application under consideration. Second, similar to most other data mining methods, the results of WGCNA can be biased or invalid when dealing with technical artifacts, tissue contamination or poor experimental design. Third, although several coexpression module detection methods are implemented within the R package, the package does not provide the means to determine which method is best. While the default hierarchical clustering methods have performed well in several real data applications, it would be desirable to compare these and other methods on a benchmark collection of real data sets. The WGCNA R package currently focuses on undirected networks. Methods for orienting edges and constructing directed networks are implemented in R, and can be used in conjunction with WGCNA [3, 8, 37].

WGCNA can be used as a data exploratory tool or as a gene screening method. For example, one may use WGCNA to explore the module structure in a network, to measure the relationship between genes and modules (module membership information), to explore the relationships between modules (eigengene networks) and to rank-order genes or modules with regard to their relationship with a sample trait. WGCNA can generate testable hypotheses for validation in independent data sets. For example, WGCNA may suggest that a module, or a putative pathway, is associated with a disease outcome. One can use a correlation test p-value [14] or a regression-based p-value for assessing the statistical significance between variables. For example, it is straightforward to attach a significance level to the fuzzy module membership metric. The relationship between standard microarray data mining techniques and gene coexpression network analysis is discussed in [22]. Coexpression networks hold great promise for uncovering the genetic basis of complex diseases, and WGCNA is an intuitive and simple tool for this task.

Acknowledgements We would like to acknowledge the grant support from 1U19AI063603-01, 5P30CA016042-28, P50CA092131, and DK072206. The authors would like to thank UCLA collaborators Jun Dong, Jake Lusis, Tom Drake, Dan Geschwind, Wen Lin, Paul Mischel, Mike Oldham, and Wei Zhao for useful discussions.

References

1. Albert, R., & Barabasi, A. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, *74*, 47–97.
2. Albert, R., Jeong, H., & Barabasi, A. L. (2000). Error and attack tolerance of complex networks. *Nature*, *406*(6794), 378–382.
3. Aten, J., Fuller, T., Lusis, A., & Horvath, S. (2008). Using genetic markers to orient the edges in quantitative trait networks: The neo software. *BMC Systems Biology*, *2*(1), 34. DOI 10.1186/1752-0509-2-34.
4. Butte, A., Tamayo, P., Slonim, D., Golub, T., & Kohane, I. (2000). Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proceedings of the National Academy of Sciences of the United States of America*, *97*, 12182–12186.
5. Cabusora, L., Sutton, E., Fulmer, A., & Forst, C. (2005). Differential network expression during drug and stress response. *Bioinformatics*, *21*(12), 2898–2905.
6. Carlson, M., Zhang, B., Fang, Z., Mischel, P., Horvath, S., & Nelson, S. F. (2006). Gene connectivity, function, and sequence conservation: Predictions from modular yeast co-expression networks. *BMC Genomics*, *7*(7), 40.
7. Carter, S., Brechb, C., Griffin, M., & Bond, A. (2004). Gene co-expression network topology provides a framework for molecular characterization of cellular state. *Bioinformatics*, *20*(14), 2242–2250.
8. Chaibub Neto, E., Ferrara, C. T., Attie, A. D., & Yandell, B. S. (2008). Inferring causal phenotype networks from segregating populations. *Genetics*, *179*(2), 1089–1100. DOI 10.1534/genetics.107.085167.
9. Chuang, H., Lee, E., Liu, Y., Lee, D., & Ideker, T. (2007). Network-based classification of breast cancer metastasis. *Molecular Systems Biology*, *3*(3), 140.
10. Clayton, D., & McKeigue, P. M. (2001). Epidemiological methods for studying genes and environmental factors in complex diseases. *Lancet*, *358*, 1356–1360.
11. Dennis, G. J., Sherman, B. T., Hosack, D. A., Yang, J., Gao, W., Lane, H. C., & Lempicki, R. A. (2003). David: Database for annotation, visualization, and integrated discovery. *Genome Biology*, *4*(5), P3.
12. Dong, J., & Horvath, S. (2007). Understanding network concepts in modules. *BMC Systems Biology*, *1*(1), 24.
13. Eisen, M., Spellman, P., Brown, P., & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, *95*(25), 14863–14868.

14. Fisher, R. A. (1915). On the ‘probable error’ of a coefficient of correlation deduced from a small sample. *Metron*, *1*, 1–32.
15. Frohlich, H., Speer, N., Poustka, A., & BeiSZbarth, T. (2007). Gosim – an r-package for computation of information theoretic go similarities between terms and gene products. *BMC Bioinformatics*, *8*(1), 166. DOI 10.1186/1471-2105-8-166.
16. Fuller, T. F., Ghazalpour, A., Aten, J. E., Drake, T. A., Lusis, A. J., & Horvath, S. (2007). Weighted gene coexpression network analysis strategies applied to mouse weight. *Mammalian Genome*, *18*(6–7), 463–472. DOI 10.1007/s00335-007-9043-3.
17. Gargalovic, P., Imura, M., Zhang, B., Gharavi, N., Clark, M., Pagnon, J., Yang, W., He, A., Truong, A., Patel, S., Nelson, S., Horvath, S., Berliner, J., Kirchgessner, T., & Lusis, A. (2006). Identification of inflammatory gene modules based on variations of human endothelial cell responses to oxidized lipids. *Proceedings of the National Academy of Sciences*, *103*(34), 12741–12746.
18. Gentleman, R., Huber, W., Carey, V., Irizarry, R., & Dudoit, S. (2005). *Bioinformatics and computational biology solutions using R and bioconductor.*. New York: Springer-Verlag.
19. Ghazalpour, A., Doss, S., Zhang, B., Wang, S., Plaisier, C., Castellanos, R., Brozell, A., Schadt, E. E., Drake, T. A., Lusis, A. J., & Horvath, S. (2006). Integrating genetic and network analysis to characterize genes related to mouse weight. *PLoS Genetics*, *2*(8), e130. DOI 10.1371/journal.pgen.0020130.
20. Han, J., Bertin, N., Hao, T., Goldberg, D., Berriz, G., Zhang, L., Dupuy, D., Walhout, A., Cusick, M., Roth, F., & Vidal, M. (2004). Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, *430*(6995), 88–93.
21. Henegar, C., Clement, K., & Zucker, J. D. (2006). Unsupervised multiple-instance learning for functional profiling of genomic data. In J. Fuernkranz, T. Scheffer, & M. Spiliopoulou (Eds.), *Machine learning: ECML 2006* (pp. 186–197). Berlin: Springer. DOI 10.1007/11871842.
22. Horvath, S., & Dong, J. (2008). Geometric interpretation of gene coexpression network analysis. *PLoS Computational Biology*, *4*(8), e1000, 117. DOI 10.1371/journal.pcbi.1000117.
23. Horvath, S., Zhang, B., Carlson, M., Lu, K., Zhu, S., Felciano, R., Laurance, M., Zhao, W., Shu, Q., Lee, Y., Scheck, A., Liau, L., Wu, H., Geschwind, D., Febbo, P., Kornblum, H., Cloughesy, T. F., Nelson, S., & Mischel, P. (2006). Analysis of oncogenic signaling networks in glioblastoma identifies aspm as a novel molecular target. *Proceedings of the National Academy of Sciences of the United States of America*, *103*(46), 17402–17407.
24. Hu, Z., Mellor, J., Wu, J., & DeLisi, C. (2004). Visant: An online visualization and analysis tool for biological interaction data. *BMC Bioinformatics*, *5*(1), Article 17.
25. Jeong, H., Mason, S., Barabasi, A., & Oltvai, Z. (2001). Lethality and centrality in protein networks. *Nature*, *411*, 41.
26. Katan, M. (1986). Apolipoprotein E isoforms, serum cholesterol, and cancer. *Lancet*, *i*, 507–508.
27. Kaufman, L., & Rousseeuw, P. (1990). *Finding rroups in data: An introduction to cluster analysis.* New York: Wiley.
28. Keller, M. P., Choi, Y., Wang, P., Belt Davis, D., Rabaglia, M. E., Oler, A. T., Stapleton, D. S., Argmann, C., Schueler, K. L., Edwards, S., Steinberg, H. A., Chaibub Neto, E., Kleinhanz, R., Turner, S., Hellerstein, M. K., Schadt, E. E., Yandell, B. S., Kendziorski, C., & Attie, A. D. (2008). A gene expression network model of type 2 diabetes links cell cycle regulation in islets with diabetes susceptibility. *Genome Research*, *18*(5), 706–716. DOI 10.1101/gr.074914.107.
29. Langfelder, P., & Horvath, S. (2007). Eigengene networks for studying the relationships between co-expression modules. *BMC Systems Biology*, *1*, 54. DOI 10.1186/1752-0509-1-54.
30. Langfelder, P., & Horvath, S. (2008). Wgcna: An r package for weighted correlation network analysis. *BMC Bioinformatics*, *9*(1), 559.
31. Langfelder, P., Zhang, B., & Horvath, S. (2007). Defining clusters from a hierarchical cluster tree: The dynamic tree cut library for R. *Bioinformatics*, *24*(5), 719–720.
32. Li, A., & Horvath, S. (2007). Network neighborhood analysis with the multi-node topological overlap measure. *Bioinformatics*, *23*(2), 222–231.
33. Little, J., & Khoury, M. J. (2003). Mendelian randomisation: A new spin or real progress? *Lancet*, *362*, 930–931.

34. Liu, M., Liberzon, A., Kong, S. W., Lai, W. R., Park, P. J., Kohane, I. S., & Kasif, S. (2007). Network-based analysis of. affected biological processes in type 2 diabetes models. *PLoS Genetics*, *3*(6), e96. DOI 10.1371/journal.pgen.0030096.
35. Oldham, M., Horvath, S., & Geschwind, D. (2006). Conservation and evolution of gene coexpression networks in human and chimpanzee brains. *Proceedings of the National Academy of Sciences of the United States of America*, *103*(47), 17973–17978.
36. Oldham, M. C., Konopka, G., Iwamoto, K., Langfelder, P., Kato, T., Horvath, S., & Geschwind, D. H. (2008). Functional organization of the transcriptome in human brain. *Nature Neuroscience*, *11*(11), 1271–1282.
37. Opgen-Rhein, R., & Strimmer, K. (2007). From correlation to causation networks: A simple approximate learning algorithm and its application to high-dimensional plant gene expression data. *BMC Systems Biology*, *1*(37).
38. Presson, A. P., Sobel, E. M., Papp, J. C., Suarez, C. J., Whistler, T., Rajeevan, M. S., Vernon, S. D., & Horvath, S. (2008). Integrated weighted gene co-expression network analysis with an application to chronic fatigue syndrome. *BMC Systems Biology*, *2*, 95. DOI 10.1186/1752-0509-2-95.
39. Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., & Barabasi, A. L. (2002). Hierarchical organization of modularity in metabolic networks. *Science*, *297*(5586), 1551–1555.
40. Reeves, W., Wagner, D., Nisenbaum, R., Jones, J., Gurbaxani, B., Solomon, L., Papanicolaou, D., Unger, E., Vernon, S., & Heim, C. (2005). Chronic fatigue syndrome-a clinically empirical approach to its definition and study. *BMC Medicine*, *3*(19).
41. Schadt, E. E., Lamb, J., Yang, X., Zhu, J., Edwards, S., GuhaThakurta, D., Sieberts, S. K., Monks, S., Reitman, M., Zhang, C., Lum, P. Y., Leonardson, A., Thieringer, R., Metzger, J. M., Yang, L., Castle, J., Zhu, H., Kash, S. F., Drake, T. A., Sachs, A., & Lusis, A. J. (2005). An integrative genomics approach to infer causal associations between gene expression and disease. *Nature Genetics*, *37*(7), 710–717.
42. Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., Amin, N., Schwikowski, B., & Ideker, T. (2003). Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research*, *13*(11), 2498–2504. DOI 10.1101/gr.1239303.
43. Shen, R., Ghosh, D., Chinnaiyan, A., & Meng, Z. (2006). Eigengene-based linear discriminant model for tumor classification using gene expression microarray data. *Bioinformatics*, *22*(21), 2635–2642.
44. Smith, G. D. (2006). Randomized by (your) god: Robust inference from an observational study design. *Journal of Epidemiology & Community Health* , *60*, 382–388.
45. Smith, G. D., & Ebrahim, S. (2003). 'Mendelian randomization': Can genetic epidemiology contribute to understanding environmental determinants of disease? *International Journal of Epidemiology*, *32*, 1–22.
46. Steffen, M., Petti, A., Aach, J., D'haeseleer, P., & Church, G. (2002). Automated modelling of signal transduction networks. *BMC Bioinformatics*, *3*(1), 34.
47. Stuart, J. M., Segal, E., Koller, D., & Kim, S. K. (2003). A gene-coexpression network for global discovery of conserved genetic modules. *Science*, *302*(5643), 249–255.
48. vanNas, A., Guhathakurta, D., Wang, S., Yehya, S., Horvath, S., Zhang, B., IngramDrake, L., Chaudhuri, G., Schadt, E., Drake, T., Arnold, A., & Lusis, A. (2008). Elucidating the role of gonadal hormones in sexually dimorphic gene co-expression networks. *Endocrinology*, *3*(150), 1235–1249.
49. Voy, B. H., Scharff, J. A., Perkins, A. D., Saxton, A. M., Borate, B., Chesler, E. J., Branstetter, L. K., & Langston, M. A. (2006). Extracting gene networks for low-dose radiation using graph theoretical algorithms. *PLoS Computational Biology*, *2*(7), e89.
50. Wei, H., Persson, S., Mehta, T., Srinivasasainagendra, V., Chen, L., Page, G., Somerville, C., & Loraine, A. (2006). Transcriptional coordination of the metabolic network in arabidopsis. *Plant Physiology*, *142*(2), 762–774.
51. Weston, D., Gunter, L., Rogers, A., & Wullschleger, S. (2008). Connecting genes, coexpression modules, and molecular signatures to environmental stress phenotypes in plants. *BMC Systems Biology*, *2*(1), 16. DOI 10.1186/1752-0509-2-16.

52. Wilcox, R. R. (2004). Introduction to robust estimation and hypothesis testing. Academic Press. ISBN:0127515429.
53. Yip, A., & Horvath, S. (2007). Gene network interconnectedness and the generalized topological overlap measure. *BMC Bioinformatics*, *8*(8), 22.
54. Zhang, B., & Horvath, S. (2005). A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology 4*(1), 17.
55. Zhou, X., Kao, M., & Wong, W. (2002). Transitive functional annotation by shortest path analysis of gene expression data. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(20), 12783–12788.

Chapter 19
Liquid Association and Related Ideas in Quantifying Changes in Correlation

Ker-Chau Li

Abstract This chapter describes a novel statistical concept of a ternary relationship between variables in a complex data system, coined 'liquid association' (LA) by Li (Proc Natl Acad Sci U S A 99(16):16875–16880, 2002). ***LA describes how variation in the pattern of association between a pair of variables, including its sign and strength, is mediated by a third variable from the background.*** LA is introduced because despite the many successful applications of similarity based analysis on microarray data, numerous cases where the functional association between genes is known from the literature (confirmed by experiments) but the statistical correlation from the corresponding expression data is practically zero also exist. Other than the noises in the microarray data, a deeper reason may be the ***biological complexity*** of the cellular system and the hidden components, which are not directly measured by gene expression, such as multiple functions of a protein, varying cellular oxidization-reduction states, fluctuating hormone levels or other cellular signals and so on.

19.1 Limitation of Correlation and Similarity Analysis

It has been well-accepted that genes with similar profiles are likely to share common cellular roles and participate in related pathways. Pearson's correlation is a popular measure of similarity, which is thought to conform well to the intuitive biological notion of what it means for two genes to be 'co-expressed' [2]. For any pair of genes (X, Y), a correlation or other similarity measure is computed. High correlation is a likely evidence for functional association between genes-their gene products are more likely to form a protein complex; they may participate in common pathways and biological processes; they may be governed by common regulatory elements upstream their gene coding regions. This kind of similarity

K.-C. Li
Institute of Statistical Science, Academia Sinica Department of Statistics, UCLA
e-mail: kcli@stat.ucla.edu

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_19,
© Springer-Verlag Berlin Heidelberg 2011

analysis together with other variations and extensions for clustering or classification has generated enormous information from high throughput biological experiments.

Positively and negatively associated genes, once detected, can be used as building blocks toward the understanding and prediction of the system behavior. But in the extremely complex biological systems, the vast majority of cases turn out showing no or weak correlations between variables. This is a major shortcoming of the traditional correlation-based similarity analysis. Even the advanced techniques like K-means, self organization maps or hierarchical clustering method do not address this issue properly.

There are numerous cases of failure with conventional correlation analysis where functional association is evident according to experimental evidence, yet the statistical correlation from gene assay evidence is practically zero. The abundance of such negative results constitutes a bottleneck in distilling more information from microarray data. Two illustrative cases concern the transcription factors, Max and thyroid hormone receptor (TR); Weaver [24], page 406–407. These transcription factors can serve either as activators or as repressors, depending on other interacting molecules. Max can bind to Myc and form a Myc-Max dimer that acts as a transcription activator. But when bound to Mad, the Mad-Max dimer serves as a repressor. For the case of TR, it associates with RXR to form a TR-RXR dimer that serves as a repressor in the absence of thyroid hormone. In the presence of thyroid hormone, the TR-RXR dimer is converted into an activator. Histone deacetylation is involved in both repressing events. Thus for example, if X is taken to be the expression profile of the gene encoding TR and Y is taken to be the profile of one of its target genes, then X and Y may be either positively correlated or negatively correlated, depending on the hormone level. If the hormone level fluctuates in an unspecific manner, the opposing directions of correlation may cancel out each other and no similarity based analysis may succeed in detecting the functional association between X and Y.

In general, all biological processes are interlocked and many proteins have multiple cellular roles. Two proteins engaged in a common process under certain conditions, may disengage and embark on activities of their own under other conditions. This implies that both the strength and the pattern of association between two gene profiles may vary as the intrinsic cellular state changes.

An important issue arising from the above discussion is how to systematically study the co-expression patterns between functionally related genes, subject to the cellular state changes. The issue is compounded by fact that a cellular state is not clearly defined and that there are numerous intracellular and intercellular conditions that can alter the cellular state. A direct approach would be to specify a number of them and conduct more profiling experiments under more specific conditions accordingly. But this depends on our biological knowledge about what conditions are relevant to the genes under study. Quite often such information is not available and indeed, biologists may hope to identify such conditions from large scale genomic data. The LA method approaches this issue in a reverse manner. For any two given genes, the method attempts to delineate the cellular state changes that may affect their co-expression pattern. This is made possible via a theory of co-expression dynamics.

19.2 Concept of Liquid Association

The term liquid association (LA), *'liquid' as opposed to 'steady'*, was first used to conceptualize the internal change of co-expression patterns for a pair of genes (X,Y) in response to constant changes in the cellular state variables. Because the relevant cellular states are typically unknown, it is difficult to detect this novel type of association from the profiles of X and Y alone. To make a progress, we make a somewhat broad assumption that the state change turns out associated with the differential expression of one gene Z. Then the profile of Z can be utilized to screen the scatterplot of (X,Y) for the intrinsic expression patterns termed LA activity. This creates a ternary relationship between X, Y and Z. Figure 19.1 is a schematic diagram for illustrating the concept of LA.

Specifically, if an increase in Z is associated with an increase in the correlation of (X,Y), then Z is a positive LA-scouting gene for (X, Y) and a positive score is assigned to quantify the strength of LA. The pair (X, Y) is called a positive LAP (liquid association pair) of Z. Likewise, a negative LA-scouting gene can be defined if an increase in Z is associated with a decrease in the correlation of (X, Y) then the LA score is negative-valued. Thus when comparing the low with the high expression levels of a positive LA-scouting gene, the scouted LAP is likely to change from being contra-expressed to being co-expressed. For a negative LA-scouting gene, the change goes the opposite direction from being co-expressed to

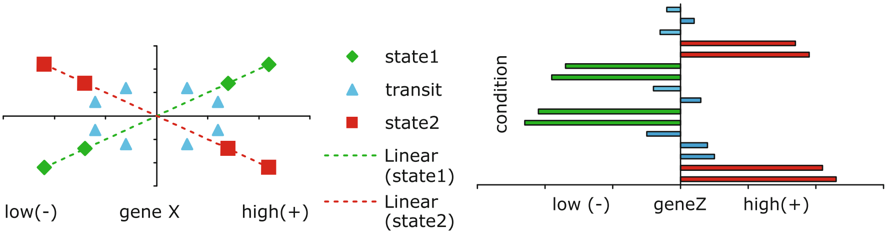


Fig. 19.1 Co-expression dynamics. Profiles of genes X and Y are displayed in a scatterplot (*the left panel*). The four green (*diamond*) points represent four conditions for cellular state 1 wherein X and Y are co-regulated. Likewise, the four red (*square*) points represent four conditions for cellular state 2 wherein X and Y are contra-expressed. To depict this kind of internal evolution in the association pattern, we say $(X\ ,Y)$ forms a liquid-association pair (LAP). Because the relevant cellular states are usually unknown, it is hard to detect LAP directly from the profiles of X and Y alone. However, if the cellular states are correlated with the differential expression of a third gene Z, then we can use Z to scout (X, Y) for information about their liquid association (LA.) activity. In the right panel, the four green bars (*low values*) represent the expression of Z for the same 4 green-colored conditions as in the left panel. Likewise, the four red bars (*high values*) correspond to the 4 red-colored conditions in the left panel. When Z is down-regulated (*green*), X and Y are co-expressed; when Z is up-regulated (*red*), X and Y become contra-expressed. We assign a score to quantify the strength of LA. The LA score for this illustration is a negative value. On the other hand, if the low expressions of Z correspond to the red points in the left panel and the high expressions of Z correspond to the green points in the left panel, then the LA score will be positive. This figure is taken from Li [10]

being contra-expressed. In general, an LA-scouting gene serves only as a red flag, a surrogate for the intrinsic state variable that facilitates the LA activity. The protein encoded by an LA-scouting gene may not have any direct physical contact with its LAP or the proteins encoded by the genes.

For the genome-wide study, there are a huge number of combinations for choosing three genes from N genes. For example, the number of combinations in yeast (with N = 5878) is approximately 33.8 billion triplets. For Affymetrix human gene expression chip U133plus2, the number of gene probes exceeds 60,000, leading to more than 3.6E12 triplets to inspect. Clearly, it is too time-consuming to visualize every scatterplot like Fig. 19.1 for detecting the LA patterns. A statistical measure to quantify liquid association for identifying the likely LA triplets is introduced next.

19.3 Mathematical Derivation

The mathematical platform for introducing liquid association involves three random variables X, Y and Z, each with mean 0 and variance 1. The correlation coefficient between X and Y is equal to $E(XY)$. By conditioning, $E(XY) = E(E(XY|Z)) = Eg(Z)$, where $g(Z) = E(XY|Z)$ denotes the conditional expectation of XY given Z. This identity describes how the variable Z contributes to the correlation between X and Y via $g(Z)$. We regard $g(z)$ as the influx of correlation contribution at $Z = z$ and ask how it varies as z increases. Denote the derivative of $g(z)$ with respect to z by $g'(z)$. We quantify the overall influx change by averaging $g'(Z)$ over Z, leading to the following definition of liquid association.

Definition 19.1. Suppose X, Y, Z are random variables with mean 0 and variance 1. The liquid association (LA) of X and Y with respect to Z is given by $LA(X, Y|Z) = Eg'(Z)$, where $g(z) = E(XY|Z = z)$.

This definition is fairly general. When Z follows a normal distribution, there is a very simple way of calculating LA.

Theorem 19.1. *If Z is standard normal, $LA(X, Y|Z) = E(XYZ)$.*

This theorem can be proved by using the celebrated Stein Lemma (Stein 1981). Using this theorem, the LA score is simply the average of the triplet product, $n^{-1}(x_1y_1z_1 + \cdots + x_ny_nz_n)$.

Remark 19.1. The notion of LA is completely different from the notion of partial correlation coefficient. The partial correlation measures how X and Y are correlated after adjusting for the common correlation due to their correlation with Z. More specifically, let $U = X - r_1Z$ and $V = Y - r_2Z$ be the residual of regressing X and Y on Z respectively; r_1, r_2 being the regression coefficients. Then the partial correlation is equal to the correlation of (U, V). One typical use of partial correlation is to help the causality inference. In some applications, two highly correlated variables X and Y may turn out having no casual relationship at all and the partial correlation technique can used to probe for a shared variable Z that is correlated with both X and Y, thus explaining the apparent correlation between them. For such cases, the partial correlation would be reduced to essentially 0.

Remark 19.2. One technical note concerns phrases such as 'change in correlation between X and Y given Z' which should be more accurately rephrased as 'change in the intrinsic structure of correlation between X and Y'. The subtlety is that $E(XY|Z = z)$ measures only the intrinsic contribution of Z to the overall correlation. It does not measure the conditional correlation of X and Y given $Z = z$, $h(z) = \rho(X,Y|Z = z) = Cov(X,Y|Z = z)/SD(X|Z = z)SD(Y|Z = z)$. One can define a liquid association-like notion using $h(z)$ to replace $g(z)$ and establish the identity with $E(h(Z)Z)$ using the Stein lemma. However, the estimation of $h(Z)$ requires local smoothing.

Remark 19.3. The estimation issue aside, on may ask which measure, $Eh'(Z)$ or $Eg'(Z)$, would be more appropriate for applications in microarray data analysis. This is not an easy question to answer. A typical use of correlation between gene expression profiles intends to reflex the degree of coordination in regulation gene expression. The quantity $g(Z)$ uses a common baseline measure for X and Y no matter what conditions are associated with Z. For many applications in microarray studies, this is easier to accept. As to be argued below, $g(z)$ reflects better the intuitive change in the co-expression/co-regulation of a pair of genes X and Y than $h(z)$.

In most microarray gene expression analysis studies [6], biologists tend to agree that the baseline expression of a gene can be represented by the average value of the expression of all conditions under study. Because we have set the mean of X and Y to zero, cases with $X > 0$ and $Y > 0$ ($X < 0$ and $Y < 0$, respectively) indicate of upregulation (down-regulation, respectively) of both genes. Co-upregulation or co-downregulation contributes a positive value in the product XY. Likewise, contra-expression/contra-regulation (either $X > 0$, $Y < 0$, or $X < 0$, $Y > 0$) contributes a negative value in the product XY. Thus by calculating the average of contribution to the product XY over all conditions, LA is a way of quantifying the change of co-regulation pattern.

The use of the conditional correlation $h(z)$, however, implies that the baseline expression levels of gene X and gene Y and their ranges of variation must be reset in assessing the degree of coordination under different activity level of Z. This mathematical formulation of conditional correlation may not serve well the purpose of explaining the intuitive biological sense of coordination under the varying conditions associated with Z. See Sun, Yuan and Li [22] for more detail discussion in the context of 2D-trait eQTL mapping in genetic variation studies.

Remark 19.4. Our LA measure is not intended to capture the detail pattern of change in $h(z) = E(XY|Z = z)$. For example, if $h(z)$ is not monotonically increasing or decreasing, then measures like $E|h'(Z)|$ may seem more informative to use.

Estimation of such measures would require local smoothing, however. Such procedures would be worth pursuing provided that good biological insight can be gained to ensure the fluctuation in $g(z)$ is not due to noises.

19.4 Computing LA

The theorem derived in the previous section makes the computation of LA extremely simple. For application in analyzing microarray gene expression data, we convert each gene expression profile by normal score transformation so that the normal distribution can be followed as closed as possible. Otherwise, to estimate LA we cannot apply Theorem and will have to resort to nonparametric regression, which would be difficult to do for all possible triplets in the genome.

The normal score transformation is $\{\Phi^{-1}(\frac{R_i}{n+1}), i = 1, 2, \ldots, n\}$, where $\Phi(\cdot)$ is the cumulative normal distribution, R_i is the rank of the gene expression for the i^{th} condition and n is the total number of conditions.

To gain some insight about the magnitude of LA score required for detecting a meaningful change in correlation, Yuan and Li [14] conducted a simulation study based on a model which was flexible enough to represent a wide range of correlation changes. They considered four independent random variables, s, Z, e_1, e_2; s taking values -1 and 1 with equal probability and Z, e_1, e_2 being standard normal random variables. Let $X = sr_1Z + (1 - r_1^2)^{1/2}e_1$ and $Y = sr_2|Z| + (1 - r_2^2)^{1/2}e_2$. It can be seen that X, Y are marginally normal with mean 0 and variance 1. For $Z < 0$, the correlation between X and Y is $-r_1r_2$ and for $Z > 0$, the correlation is changed to r_1r_2. By setting $r_1 = r_2 = 0.6$ and generating three gene profiles with 60 cases: $x_1, \ldots, x_{60}$; $y_1, \ldots, y_{60}$; $z_1, \ldots, z_{60}$ to compute the LA scores for 10,000 repeats, they obtained the mean of LA scores to be .316 and the standard deviation 0.123. The simulation for this and other correlation changes were shown in the following table. Using 95% as the cutoff point, we see that a change from -0.5 to 0.5 can be comfortably detected by LA. In fact, only 1.22% of cases simulated under no correlation change ($r_1 = r_2 = 0$) have LA scores higher than 0.2389(the average LA score for $r_1 = r_2 = 0.5$); see the last row of the Table 19.1. To study the effect of the number of conditions, they also conducted the simulation for 40 conditions. The results are reported in parentheses. The change of -0.5 to 0.5 may still be detectable.

For easy visualization of LA activity, it helps to find good threshold values c_1, c_2 for dividing conditions into the high ($Z > c_2$) , the intermediate ($c_1 < Z < c_2$) and the low ($Z < c_1$) expression groups of the LA-scouting gene Z. There are at least two ways to proceed. The first method uses a stratified linear model as an approximation to fit the data by least squares. The model takes the form of $Y = a_1 + b_1X + \mathit{error}$ for $Z < c_1$ and $Y = a_2 + b_2X + \mathit{error}$ for $Z > c_2$, where a_1, b_1, a_2, b_2, c_1 and c_2 are estimated from the data. The second method based on the optimal accumulation of correlation contribution by Z so that the total contribution

Table 19.1 Rows 1–4 show the results of simulation under each specified value of $r_1 = r_2$. Row 5 shows the result simulated under $r_1 = r_2 = 0$. The number of conditions is set at 60 (40 for values reported in parentheses)

$r_1 = r_2 =$	0	0.3	0.4	0.5	0.6	0.7	0.8	0.9
mean of	.0000	.0993	0.1653	0.2389	.3159	.3920	0.4648	0.5336
LA score	(.000)	(.0879)	(.1489)	(.2151)	(.2831)	(.3537)	(.4186)	(.4793)
S.D. of LA	.1072	0.1171	0.1187	0.1215	.1226	0.121	0.1174	0.1139
score	(.1241)	(.1331)	(.1367)	(.1369)	(0.1367)	(0.1375)	(.1339)	(0.1297)
% of LA	50.24%	80.46%	91.76%	97.28%	99.42%	99.87%	99.97%	100%
score > 0	(50.59%)	(74.75%)	(86.08%)	(94.04%)	(97.64%)	(99.20%)	(99.74%)	(99.94%)
% of LA score >	50.24%	17.76%	6.24%	1.22%	0.17%	0%	0%	0%
mean of LA score	(50.59%)	(24.11%)	(11.53%)	(4.03%)	(1.14%)	(0.19%)	(0.06%)	(0.01%)

to the product XY will be of different signs and the difference will be as large as possible between the high and the low expression groups of gene Z.

19.5 Examples of Application

Example 1. Gene regulation for the metabolic pathway of urea cycle. This example is taken from Li [10]. In Fig. 19.2, a schematic of the urea cycle with key enzymes and intermediates is shown. The biological function of this pathway is to maintain a suitable level of arginine. The expression data we used came from four cell-cycle experiments, accessible at http://genome-www.stanford.edu/cellcycle. All of the data were used to construct gene profiles with a total of 73 conditions.

ARG2 is the gene encoding acetylglutamate synthase which carries out the first step in synthesizing ornithine and eventually arginine. In order to feed ornithine into the arginine biosynthesis pathway, CAR2 (ornithine aminotransferase) should be inactivated to avoid the immediate degradation of ornithine. This suggests that CAR2 and ARG2 may be contra-expressed. However, the correlation between ARG2 and CAR2 is nearly zero.

To apply liquid association, we take genes (ARG2, CAR2) as the gene pair (X, Y) to calculate $LA(X, Y|Z)$ for each Z of the 5,878 genes in the database and rank the results. From the list of 10 genes with most negative LA scores, we found the gene CPA2 which encodes the large subunit of carbamoyl phosphate synthetase. Because carbamoyl phosphate is needed for enzyme ornithine transcarbamoylase (encoded by ARG3) to synthesize citrulline from ornithine, high expression of CPA2 reflects the state of cellular demand for arginine.

The LA activity pattern for (ARG2,CAR2) as mediated by CPA2 is shown in Fig. 19.3. It can be seen that when the expression level of CPA2 is low, as represented by the diamond shapes, a positive correlation is seen between ARG2 and CAR2. As the level of CPA2 increases, the correlation pattern is gradually weak-

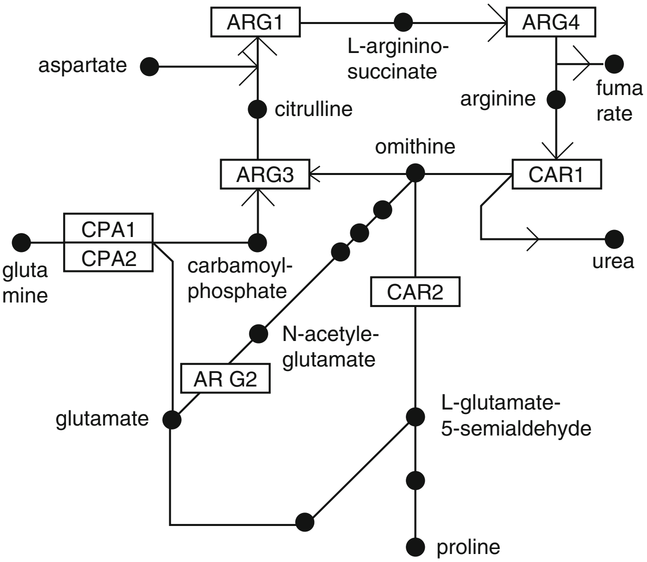


Fig. 19.2 Urea cycle/arginine biosynthesis pathway. ARG2 encodes acetyl-glutamate synthase, which catalyzes the first step in synthesizing ornithine from glutamate. Ornithine and carbamoyl phosphate are the substrates of the enzyme ornithine transcarbamoylase, encoded by ARG3. Carbamoyl phosphate synthetase is encoded by CPA1 and CPA2. ARG1 encodes argininosuccinate synthetase, ARG4 encodes argininosuccinase, CAR1 encodes arginase, and CAR2 encodes ornithine aminotransferase. This figure is taken from Li [10]

ened. Eventually, when CPA2 is high, shown as triangles, the association has turned into a negative. The LA score is −0.2894 with the P-value 56 of a million by a permutation test as described in Li [10].

For efficient activation of the arginine biosynthesis pathway, up-regulation of ARG2 must be concomitant with down-regulation of CAR2 to prevent immediate ornithine degradation. We see this occurs only when CPA2 is up-regulated. Because activation of CPA2 provides the influx of carbamoyl phosphate into urea cycle, high expression level of CPA2 can be interpreted as a physiological signal for arginine demand. Therefore, from the LA-activity plot it can be seen that under this state, *ARG2* and *CAR2* are indeed *negatively* correlated. When the demand is relieved and CPA2 is lowered, CAR2 is up-regulated, opening up the channel for ornithine to degrade and leave the urea cycle.

Example 2 Gene expression and drug sensivity. Microarrays have been applied to the 60 human cancer cell lines (9 tumor types: 8 breast, 6 central nervous system, 7 colon, 6 leukemia, 8 melanoma, 9 non-small cell lung cancer, 6 ovary, 2 prostate, 8 renal) used in the NCI drug discovery screen [16,18,19]. Genes whose expression correlates well with a drug activity profile are likely to be associated with the underlying cellular mechanism of growth inhibition. However, there are numerous factors

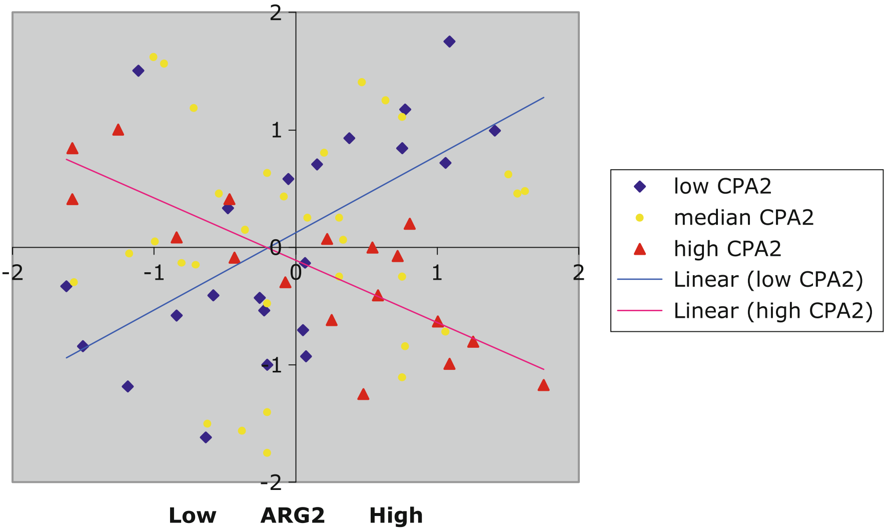


Fig. 19.3 Liquid association between ARG2 and CAR2 as mediated by CPA2. When the expression level of CPA2 is low (conditions represented by blue diamonds), a positive correlation is seen between ARG2 and CAR2. As the level of CPA2 increases, the correlation pattern is gradually weakened. Eventually, when CPA2 is high (*red triangle*), the association is turned into negative. The LA score is −0.289. For efficient activation of the arginine biosynthesis pathway, up-regulation of ARG2 must be concomitant with down-regulation of CAR2 to prevent ornithine from leaking out of the urea cycle. We see this occurs only when CPA2 is up regulated. Because activation of CPA2 provides the influx of carbamoyl phosphate into urea cycle, high expression level of CPA2 can be interpreted as a physiological signal for arginine demand. When the demand is relieved and CPA2 is lowered, CAR2 is up-regulated, opening up the channel for ornithine to leave the urea cycle. This figure is taken from Li [10]

that can contribute to drug sensitivity. Consequently, quite often no correlation is found between a drug activity profile and the expression profile of its known molecular targets. Indeed, extensive biochemical studies have been conducted on issues such as drug transport, modification, translation regulation of the target gene, cell cycle arrest, and programmed death; see for example, Chapter 19, pharmacology of cancer chemotherapy, of Devita, Hellman and Rosenberg [5]. But the drug resistance problem is complex and it would be useful to find a computational method for augmenting the sketchy results established by the labor intensive biochemical approaches. The LA method offers one such approach, because of its ability to exploit lack of correlation.

Li and Yuan [14] investigated how to correlate drug activity profiles with gene expression profiles. This is achieved by putting both the gene expression profiles and the drug activity profiles together as an enlarged system. The refined database consists of indices of $LA(X, Y|Z)$, where X, Y, Z are any drug activity profile or gene profile. The variety of gene- drug combination offers flexibility in using the refined database that can be specified at the user preference/interface stage. For example, if X is the activity profile of a drug of interest, Y is a drug target gene profile and Z is any other gene profile, then by comparing $L(X, Y|Z)$ over Z, the high score genes may have a role in affecting the drug activity.

The use of antifolate compounds in chemotherapy have a long history in medicine. Methotrexate(MTX) is an antifolate that has been used for treatment of non-Hodgkin's lymphoma, osteogenic sarcoma, chorocarcinoma and carcinomas of breast, head and neck. The molecular target of MTX is DHFR (dihydrofolate reductase). But the correlation between MTX and DHFR and that between MTX and TYMS are weak. We took X = MTX (Methotrexate), Y = DHFR (dihydrofolate reductase). It was found that TXN(thioredoxin) has the fourth highest negative LA score. This signifies a central role for the thiol-disulfide redox regulation in tumor growth and drug resistance and is consistent with the view that the thioreduction system plays a role in all three major aspects with which the clinician is concerned: prevention, early detection and effective treatment. Furthermore, from the short list of the significant LA-scouting genes for (MTX, DHFR) and that for (MTX, TYMS), TXN (thioredoxin) was also found. In addition, the list also included TXNRD1 (thioredoxin reductase).

Example 3 Alzheimer Disease. Amyloid-beta peptide is the predominant component of senile plagues in the brains of patients with Alzheimer's disease (AD). It is derived from the amyloid-beta precursor protein (APP) via the consecutive **proteolytic cleavage by beta secretase** at the N terminus **and by gamma secretase** at the C terminus. APP is a widely-expressed cell surface protein. Its normal role was first linked to the control of gene expression in Cao and Südhof [3], where the carboxyl-terminal intracellular fragment of APP was found to interact with the nuclear adaptor protein Fe65 (encoded by APBB1) and the histone acetyltransferase Tip60 (encoded by HTATIP). In Li et. al. [14], we compare the profiles of APBB1 and HTATIP with that of APP and find the correlations (−.06, −.27 respectively) are quite low. In search of genes, which may play a role in weakening the correlation, we first apply LA to the pair (APP, PBB1). We find a **beta-site APP-cleaving enzyme BACE2** from the best 20 genes with negative LA scores. We then apply LA again to the pair (APP, HTATIP). This time we find a major component of **gamma-secretase PSEN1** (presenilin 1) to be at the second place with best positive LA scores! There are several other high LA score genes that are related to Alzheimer diseases; see Li et al. [11] for details.

Example 4. Multiple sclerosis (MS). This example was taken from Li et al. [12]. To test the applicability of LA in the characterization of the molecular background of a complex trait, we selected available information existing for MS in a special population sample of Finland. We started with the major MS candidate gene MBP(myelin basic protein, compacting and stabilizing myelin sheath). The role of this gene has been proven in rodent models for MS, EAE (experimental allergic encephalomyelitis), and the gene has been identified both in linkage and in association study in large MS pedigrees of a regional sub-isolate of Finland. Over the years, geneticists have conducted extensive MS studies using the Finnish population and identified three major genetic loci for familial MS in this isolated population: HLA on 6p, MBP on 18q and loci on 17q22–24 and 5p12–p14. A recent work gives a refined MS locus of 2.5 Mb on chromosome 17q22–q24 and establishes PRKCA (protein kinase C, alpha) as a primary candidate gene in this region. Involvement of this gene with MS is further validated by an association with MS in a UK population. The protein kinase C isoforms are involved in pathways regulating a large number of cellular processes such as proliferation, apoptosis, differentiation, migration, and neuronal signaling.

To study the co-expression pattern between MBP and PRKCA, we take them as genes X and Y to explore GNF2002 database [20] through the LA system. The gene with the highest LA score is SLC1A3 (glial high glutamate transporter, member 3). Interestingly, SLC1A3 is located near the boundary of the MS locus on 5p. Subsequent liquid association analyses lead to many other functionally associated genes, using four databases in total, human tissue data of GNF2002 and GNF2004 [21] and NCI cell line data of NCI_Affy [19] and NCI_cDNA [16]. Figure 19.4 summarizes the key findings. It can be seen that the HLA locus is also associated with SLC1A3 and MS, independently of our previous knowledge of this well established fact.

A follow-up gene typing study on Finnish MS families was conducted and established SLC1A3 as a candidate gene of MS. Moreover, stratification of the Finnish

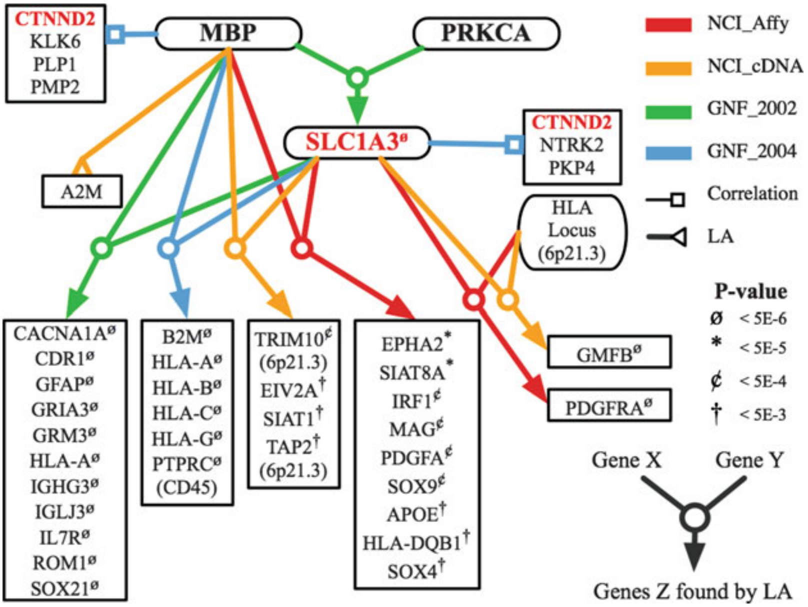


Fig. 19.4 SLC1A3 and related genes. Four large scale gene expression databases are used in this study. The arrows point to the genes found by the liquid association score system. The color of a line/arrow shows which database is used in the analysis. P-values are calculated by randomization test. All four major MS loci for the Finnish scan have representative genes in this chart: MBP from 18q23, PRKCA from 17q22–q23.2, SLC1A3 from 5p13, and the HLA locus at 6p21.3. Also shown are two separate lists of genes correlated with MBP and with SLC1A3 most strongly. CTNND2 (located at 5p15.2) is seen in both lists. This figure is taken from Li et al. [12]

families according to HLA type based on the overtransmission of high-risk MS DR2 allele belonging to HLA region to MS-affected individuals from our study sample further strengthened the resulting association between the SLC1A3 SNPs and MS. Specifically, TDT for the SNP rs2562582 (located hear SLC1A3) showed significant association with MS, p-value 0.0006. Thus, based on the LA and further supported by association analyses, SLC1A3 connects all four major MS loci identified in Finnish families.

The findings by LA analysis were strengthened with results from a more recent international MS Whole Genome Association scan [7]. A major component of the study used Affymetrix 500 K to screen common genetic variants of 931 MS family trios. Based on their data released, two SNPs, rs4869676 (chromosome 5: 36641766) and rs4869675 (chromosome 5: 36636676), with TDT P values of 0.0221 and 0.00399 respectively, were found in the upstream regulatory region of the SLC1A3 gene. In fact, within the 1 Mb region of rs486975 there are a total 206 SNPs in the Affymetrix 500 K chip. No other SNPs have P values less than that of rs486975.The next most significant SNPs in this region are rs1343692(chromosome 5: 35860930) and rs6897932 (chromosome 5: 35910332; the identified MS susceptibility SNP in the IL7R axon). The MS marker we identified, rs2562582(chromosome 5: 36641117), less than 5 kilobases away fromrs4869675, was not used in the Affymetrix chip. Interesting, IL7R also appeared in the genes list found by our LA analysis; see Fig. 19.4, the leftmost panel of genes which were found by setting X,Y to be MBP,SLC1A3 to search for LA scouting genes.

19.6 A Higher Dimension Generalization of LA

While LA is motivated by studying the intrinsic change in correlation structure between two genes, it is natural to ask how to deal with multiple genes. Li et al [14] introduced projective liquid association (PLA). Like LA, PLA assigns a score PLA$(X|Z)$ to a group X of variables for assessing change in the correlation structure mediated by the variable Z.

Let $X = (x_1, \ldots, x_p)'$, denote a vector of p variables, each variable being the expression of one gene in the group. To project X to a two dimensional space, we need two orthogonal vectors, a, b, $a'b = 0$. The liquid association between $a'X$ and $b'X$ as mediated by Z is given by $LA(a'X, b'X|Z) = a'E(ZXX')b$. Thus the most informative projection for revealing the LA pattern can be found by maximizing $|a'E(ZXX')b|$. Li et al. [14] showed that the solution is the difference between the largest and the smallest eigenvalues from the eigenvalue decomposition of the matrix $E(ZXX')$:

$$E(ZXX')b_i = \lambda_i b_i, \quad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$$

The solution $\lambda_1 - \lambda_p$ is achieved by taking $a = (b_1 + b_p)/\sqrt{2}$ and $b = (b_1 - b_p)/\sqrt{2}$.

This eigenvalue decomposition solution looks similar to a regression dimension reduction method, principal Hessian direction (PHD), which uses the following eigenvalue decomposition:

$$\Sigma_{yxx} b_i = \lambda_i \Sigma_x b_i \quad |\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_p|$$

where the matrix Σ_{yxx} is equal to $E((Y - \mu)XX')$, $EY = \mu$. Li [9] showed that if X follows a multivariate normal distribution, then under the dimension reduction model that the response variable Y depends on the regressor X through a k-dimensional projection of X, at most k eigenvalues of PHD eigenvalue decomposition are nonzero.

19.7 How to Decide if a LA Score is Significant or not.

In Li [10], a permutation test is proposed to serve for this purpose . Fixing the gene pair (X, Y), the procedure generates as many as 105 or 106 artificial profiles Z^* by randomly permuting the coordinate of the expression profile of gene Z and compute their LA scores LA$(X, Y|Z^*)$. This generates the null distribution. The p-value for LA$(X, Y|Z)$ is obtained by counting how often LA$(X, Y|Z^*)$ exceeds LA$(X, Y|Z)$.

An immediate question arises: how to adjust for multiple testing? Many multiple hypothesis testing procedures have been proposed for controlling family-wise error rate (FWER) or the false discovery rate (FDR) at a pre-specified level. One major difficulty in FDR procedures is the assumption of independence on the test statistics. Because in our problem, the test statistics are highly dependent, alternative solutions would be desirable. One possibility is to use an idea similar to Westfall, Zaykin and Young [25] as outlined before . However, the computation is very intensive.

For a given pair (X, Y) of genes, LA$(X, Y|Z)$ is computed for every gene Z in the genome. But practically, the most interesting ones are from the short list of genes with the most positive or the most negative scores. We like to test if these high score genes are significant or not. The idea is again to use a randomization test. But instead of permuting the coordinate of the expression profile of gene Z, we permute the coordinate of the expression profile of the gene pair (X, Y) to generate an artificial pair (X^*, Y^*). We compute LA$(X^*, Y^*|Z)$ for every Z in the genome. We then rank these simulated LA scores and keep track of the highest value (most positive) and the lowest value (most negative). After doing this several thousand times, we obtain a reference distribution for the highest LA score and a reference distributions for the lowest LA score. We can use these two distribution to obtain a P-value for the gene with highest LA score and a P-value for the gene with the lowest LA score. Similarly, if we keep track of the second highest score during the simulation, we can generate a reference distribution to obtain a P-value for the gene with the second highest LA score, and so on.

19.8 Extension for Applications with Censored Data.

One important clinical application of microarray technology is to predict survival time based on the gene expression profile. While formally the survival time is just another external variable, like the drug response profile, we often need to deal with severe censoring problem. For example, in the breast cancer data of van de Vijver [23], there were 216 right-censored patients out of the 295 cases studied. For these patients, one only knew that their times to event were greater than the last follow-up time. A direct application of LA would lead to unbiased results. However, it is possible to modify LA by working on properly imputed survival times.

A two-step procedure. In the first step, we may use Kaplan-Meier estimator to yield a crude estimate of the survival time for censored patients. With the corrected survival time as the lead, one may apply both correlation and LA analysis to find a short list of high score genes that are associated with the survival outcome. In the second step, one may apply the modified SIR for censored data [13] to give a refined estimate of the true survival time for the censored cases. Wu et al. [26] illustrated how to implement such a procedure for the aforementioned breast cancer data.

19.9 Incorporation of Discrete Variables such as Genetic Markers

LA was primarily motivated from the consideration that the cellular state can be modeled as a continuous variable. However, one may apply the same concept to the cases with discrete states. The simplest case is when Z is binary, with a notable application on the identification of genetic markers associated with complex diseases or traits. For example, in studying the gene-environment interaction, one may take X = trait of interest, Y = environment variable and search for marker Z with best LA scores. Yet another application of LA can be found in identifying marker–marker interaction. By taking Z = trait of interest, we can conduct a genome-wide search of marker pairs Y,Z with most significant LA scores.

In addition, many studies have shown that gene expression variation is heritable. The eQTL approach expands the traditional genetic study on the identification of the gene or genes directly responsible for a phenotype variation by treating the expression of a gene as a quantitative trait. This approach has been applied in yeast, mouse, rat and human [1, 4, 8, 15, 17, 27]. Taking one step further, one asks how the genetic variation may affect the co-expression pattern of a pair of genes. This is in line with the basic idea of LA. The mouse, rat and human marker profiles are more complicated because we have to differentiate homozygous genotypes from heterozygous genotypes.

There are several variants of conducting LA type of analysis when Z is a categorical variable. If Z is binary (say, taking 0 or 1), an obvious version of LA is $E(XY|Z = 1) - E(XY|Z = 0)$. This is equivalent to the basic LA formula $E(XYZ)$ if instead of using 0 and 1 to code Z, $1/a$ and $-1/b$ are used, where $a = P(Z = 1), b = 1 - a$. But this coding gives $var(Z) = 1/ab$. Thus it tends to assign higher LA scores to unbalanced markers (a or b closer to 0). To achieve equal variance $var(Z) = 1$, we may use $(b/a)^{1/2}$ and $-(a/b)^{1/2}$ for coding Z. For an application of LA in yeast eQTL study, see Sun et al. [22].

If Z has more than two categories, there are even more possibilities. For the mouse marker profiles, there are least three different types of coding, depending on the dominant/recessive considerations: $Z = -1, 0, 1$ (homozygous I, heterozygous, homozygous II); $Z = -1, 1$ (homozygous I, heterozygous or homozygous II); $Z = -1, 1$ (homozygous I or heterozygous, homozygous II). We may also consider the variance adjustment as in the binary case.

For other situations where all categories are treated equally, one possibility is to use the maximum from all binary differences, $E(XY|Z = i) - E(XY|Z = j)$, where i, j are any two values that Z may take. Another possibility is to consider the one-way analysis of variance by treating XY as the output and Z as group indicator. This leads to the quantity, $var(E(XY|Z))$.

References

1. Brem, R., Yvert, G., Clinton, R., & Kruglyak, L. (2002). Genetic dissection of transcriptional regulation in budding yeast. *Science*, *296*, 752–755.
2. Brown, P. O., & Botstein, D. (1999). Exploring the new world of the genome with DNA microarrays. *Nature Genetics*, *21*, 33–37.
3. Cao, X., & Südhof, T. C. (2001). A transcriptionally [correction of transcriptively] active complex of app with fe65 and histone acetyltransferase tip60. *Science*, *293*(5527), 115–120.
4. Chesler, E. J., Lu, L., Shou, S., Qu, Y., Gu, J., Wang, J., et al. (2005). Complex trait analysis of gene expression uncovers polygenic and pleiotropic networks that modulate nervous system function. *Nature Genetics*, *37*(3), 233–242.
5. Devita, V., Hellman, S., & Rosenberg, S. (2001). *Cancer: Principles and practice of oncoloy* (6th ed.). Philadelphia, PA: Lippincott Williams & Wilkins.
6. Eisen, M. B., Spellman, P. T., Brown, P. O., & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, *95*(25), 14863–14868.
7. Hafler, D. A., Compston, A., Sawcer, S., Lander, E. S., Daly, M. J., De Jager, P. L., et al. (2007). Risk alleles for multiple sclerosis identified by a genomewide study. *The New England Journal of Medicine*, *357*(9), 851–862.
8. Hubner, N., Wallace, C. A., Zimdahl, H., Petretto, E., Schulz, H., Maciver, F., et al. (2005). Integrated transcriptional profiling and linkage analysis for identification of genes underlying disease. *Nature Genetics*, *37*(3), 243–253.
9. Li, K. C. (1992). On principal Hessian directions for data visualization and dimension reduction: Another application of Stein's lemma. *Journal of the American Statistical Association*, *87*, 1025–1039.

10. Li, K. C. (2002). Genome-wide coexpression dynamics: Theory and application. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(16), 16875–16880.
11. Li, K. C., Liu, C. T., Sun, W., Yuan, S., & Yu, T. (2004). A system for enhancing genome-wide coexpression dynamics study. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(44), 15561.
12. Li, K. C., Palotie, A., Yuan, S., Bronnikov, D., Chen, D., Wei, X., et al. (2007). Finding disease candidate genes by liquid association. *Genome Biology*, *8*(10), R205.
13. Li, K. C., Wang, J. L., & Chen, C. H. (1999). Dimension reduction for censored regression data. *The Annals of Statistics*, *27*, 1–13.
14. Li, K. C., & Yuan, S. (2004). A functional genomic study on nci's anticancer drug screen. *The Pharmacogenomics Journal*, *4*, 127–135.
15. Morley, M., Molony, C. M., Weber, T. M., Devlin, J. L., Ewens, K. G., Spielman, R. S., et al. (2004). Genetic analysis of genome-wide variation in human gene expression. *Nature*, *430*(7001), 743–747.
16. Ross, D. T., Scherf, U., Eisen, M. B., Perou, C. M., Rees, C., Spellman, P., et al. (2000). Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics*, *24*(3), 227–235.
17. Schadt, E. E., Monks, S. A., Drake, T. A., Lusis, A. J., Che, N., Colinayo, V., et al. (2003). Genetics of gene expression surveyed in maize, mouse and man. *Nature*, *422*(6929), 297–302.
18. Scherf, U., Ross, D. T., Waltham, M., Smith, L. H., Lee, J. K., Tanabe, L., et al. (2000). A gene expression database for the molecular pharmacology of cancer. *Nature Genetics*, *24*(3), 236–244.
19. Staunton, J. E., Slonim, D. K., Coller, H. A., Tamayo, P., Angelo, M. J., Park, J., et al. (2001). Chemosensitivity prediction by transcriptional profiling. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(19), 10787–10792.
20. Su, A. I., Cooke, M. P., Ching, K. A., Hakak, Y., Walker, J. R., Wiltshire, T., et al. (2002). Large-scale analysis of the human and mouse transcriptomes. *Proceedings of the National Academy of Sciences*, *99*(7), 4465–4470.
21. Su, A. I., Wiltshire, T., Batalov, S., Lapp, H., Ching, K.A., Block, D., et al. (2004). A gene atlas of the mouse and human protein-encoding transcriptomes. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(16), 6062.
22. Sun, W., Yuan, S., & Li, K. C. (2008). Trait-trait dynamic interaction: 2D-trait eQTL mapping for genetic variation study. *BMC Genomics*, *9*, 242.
23. van de Vijver, M. J., He, Y. D., van't Veer, L. J., Dai, H., Hart, A. A. M., Voskuil, D. W., et al. (2002). A gene-expression signature as a predictor of survival in breast cancer. *The New England Journal of Medicine*, *347*(25), 1999–2009.
24. Weaver, R. (2002). *Molecular biology*. New York: McGraw-Hill.
25. Westfall, P. H., Zaykin, D. V., & Young, S. S. (2002). *Biostatistical methods* (Vol. 184, pp. 143–168). chap. Multiple tests for genetic effects in association studies. Totowa, NJ: Humana.
26. Wu, T., Sun, W., Yuan, S., Chen, C. H., & Li, K. C. (2008). A Method for Analyzing Censored survival phenotype with gene expression data. *BMC Bioinformatics*, *9*, 417. DOI 10.1186/1471-2105-9-417
27. Yvert, G., Brem, R. B., Whittle, J., Akey, J. M., Foss, E., Smith, E. N., et al. (2003). Trans-acting regulatory variation in Saccharomyces cerevisiae and the role of transcription factors. *Nature Genetics*, *35*(1), 57–64.

Chapter 20
Boolean Networks

Tung-Hung Chueh and Henry Horng-Shing Lu

Abstract Reconstruction of genetic regulatory networks from gene expression profiles and protein interaction data is a critical problem in systems biology. Boolean networks and their variants have been used for network reconstruction problems due to Boolean networks' simplicity. In the graph of a Boolean network, nodes represent the statuses of genes while the edges represent relationships between genes. In a Boolean network model, the status of a gene is quantized as 'on' or 'off', representing the gene as being 'active' or 'inactive' respectively. In this chapter, we will introduce the basic definitions of Boolean networks and the analysis of their properties. We will also discuss a related model called probabilistic Boolean network, which extends Boolean networks in order to have the advantage of modeling with data uncertainty and model selection. Furthermore, we will also introduce directed acyclic Boolean network and the statistical method of SPAN to reconstruct Boolean networks from noisy array data by assigning an s-p-score for every pair of genes. At last, we will suggest possible directions for future developments on Boolean networks.

20.1 Introduction

In order to understand complex biological networks and systems biology pathways, we need to investigate global structures instead of individual behaviors since there are interactions and associations between genes. Due to the invention of high throughput technology, genome-wide expression profiles can be measured simultaneously. However, it is still a great challenge to reconstruct functional network

T.-H. Chueh
Institute of Statistics, National Chiao Tung University, Hsinchu, Taiwan, Republic of China
e-mail: u9126802@stat.nctu.edu.tw

H. Horng-Shing Lu (✉)
Institute of Statistics, National Chiao Tung University, Hsinchu, Taiwan, Republic of China
e-mail: hslu@stat.nctu.edu.tw

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_20,

© Springer-Verlag Berlin Heidelberg 2011

architectures and to identify complex biological networks from genomewide data such as DNA sequences and expression profiles, because the number of gene interactions is huge [4]. In recent years, there has been significant progress in research and development concerning genetic network models and network reconstruction problems.

Various methods have been proposed in the literature to tackle the problem reconstructing of genetic regulatory networks. For instance, the Bayesian network model is an important technique that has been studied extensively in the past 2 decades [12, 13, 22, 23]. A Bayesian network is a directed acyclic graph (DAG) comprised of two components: the first component is comprised of nodes that correspond to a set of variables and a set of directed edges between variables with Markov properties. The second component describes a conditional distribution for each variable, given its parents in the graph. Recently, Bayesian network models have been applied to analyze microarray expressions and biological data [6, 10, 11]. Although algorithms for reconstructing Bayesian networks have been developed [8, 32], the algorithms' computational costs remain a concern as Bayesian networks with a small number of variables still require large sample sizes in order to obtain accurate estimates.

Therefore, we consider a simpler model: Boolean networks, which can be represented as binary and switching biological networks. Boolean networks were originally introduced by Kauffman in 1969 [14], and received attention in the studies of gene regulatory networks because of Boolean networks' simple structures. We regard Boolean networks as a generalization of Boolean cellular automata (CA) where the state of each node is affected by other nodes in the network [36, 37]. In Boolean network models, nodes represent the statuses of genes and gene expression states are quantized to one of two states: on or off, representing a gene as active or inactive, respectively. The wiring with rules between nodes in the graph represents a functional link between genes and determines the expressions of target genes given a series of input genes. Hence, the target gene is influenced by a set of genes with specific rules under the structure of Boolean networks.

Classical Boolean networks have been criticized for their deterministic nature. The assumption that every gene is determined only by a single Boolean function with a fixed set of input genes may be unsound. Therefore, we will discuss a more flexible model in the literature called probabilistic Boolean networks [26, 27], which allow more than one Boolean function for every target gene. The probabilistic Boolean network model can handle uncertainty in data and model selection, but still retain the exquisite rule-based properties of Boolean networks. Further, we will discuss directed acyclic Boolean networks and the statistical method of SPAN [20] to infer pairwise relationships and reconstruct Boolean networks from noisy array data by assigning a s-p-score for every pair of genes.

This chapter is organized as following: In Sect. 20.2, we will introduce the definition of classical Boolean networks and discuss the network reconstruction algorithm from input/output profiles of gene expression along with the computational complexity and the number of experiments required for inferring the classical Boolean networks [1, 3, 14, 15]. In Sect. 20.3, we will focus on the definition and properties of probabilistic Boolean networks which generalize classical Boolean networks. We will also discuss the inference of probabilistic Boolean networks [26]. Then, we will introduce directed acyclic Boolean networks and the statistical reconstruction method of SPAN by considering the pairwise relationships of every elements [20] in Sect. 20.4. Lastly, we will conclude and discuss future developments on Boolean networks in Sect. 20.5.

20.2 Boolean Networks

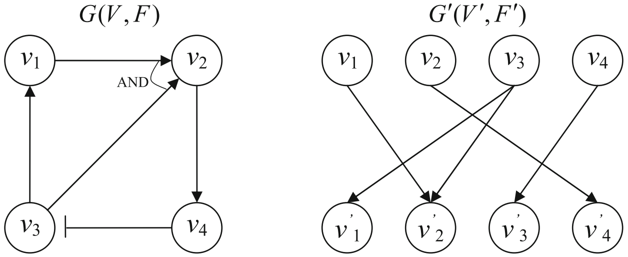
Boolean networks (also known as random Boolean networks or N-K models) were introduced 30 years ago by Kauffman to represent genetic regulatory networks [14]. In this section, we will review the definition of Boolean networks and introduce the network reconstruction algorithm from state transition tables that are related to profiles of gene expression. We will also discuss the computational complexity and the amount of data required for the reconstruction of a network structure.

20.2.1 Definition of Boolean Networks

A Boolean network $G(V, F)$ is a directed graph consisting of two components: a set of nodes $V = \{v_1, v_2, \ldots, v_n\}$ that corresponds to genes, and a list of Boolean functions $F = \{f_1, f_2, \ldots, f_n\}$ that corresponds to the rule of interaction and combination of several genes [2]. For every node $v_i \in V$, its expression has only two states, on and off, representing whether a gene is active or inactive. For every Boolean function $f_i(v_{i_1}, v_{i_2}, \ldots, v_{i_k}) \in F$, k specified input nodes $v_{i_1}, v_{i_2}, \ldots, v_{i_k}$ are assigned to the node v_i in the graph and represent the rules of regulatory mechanisms between genes. The expression of a gene is determined by the expression of the gene directly affecting it with a Boolean function. Therefore, the state of each node $v_i \in V$ is determined by the Boolean function $f_i(v_{i_1}, v_{i_2}, \ldots, v_{i_k})$.

For each node v_i, the gene expression state at time t is assumed to take either 0 (not-expressed) or 1 (expressed) and is expressed as $\psi_t(v_i)$. In a classical Boolean network, every gene expression profile at time $t + 1$ is completely determined by the expression profile of a set of genes $v_{i_1}, v_{i_2}, \ldots, v_{i_k}$ at time t and the corresponding Boolean function $f_i \in F$. That is, we can write $\psi_{t+1}(v_i) = f_i(\psi_t(v_{i_1}), \psi_t(v_{i_2}), \ldots, \psi_t(v_{i_k}))$.

For convenience, we converted the classical Boolean network $G(V, F)$ to the wiring diagram $G'(V', F')$ (See Fig. 20.1) [30]. For each node $v_i \in V$, suppose $v_{i_1}, v_{i_2}, \ldots, v_{i_k}$ are the input nodes assigned to v_i. Then we construct an additional node v'_i and connected the edge from v_{i_j} to v'_i for each $1 \leq j \leq k$. That is, the set of $\{v_1, \ldots, v_n\}$ represents the gene expression profile at time t and the set of $\{v'_1, \ldots, v'_n\}$ corresponds to the gene expression profile at time $t + 1$. Hence we can treat the set of $\{v_1, \ldots, v_n\}$ as the input values and the set of $\{v'_1, \ldots, v'_n\}$ as the corresponding output values. Therefore, the output values of $\{v'_1, \ldots, v'_n\}$ are determined by $v'_i = f_i(v_{i_1}, \ldots, v_{i_k})$.



$$v_1' = v_3 \quad v_2' = v_1 \text{ AND } v_3 \quad v_3' = \text{NOT } v_4 \quad v_4' = v_2$$

INPUT	v_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
	v_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
	v_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	v_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
OUTPUT	v_1'	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
	v_2'	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
	v_3'	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
	v_4'	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

Fig. 20.1 A Boolean network $G(V, F)$, its wiring diagram $G'(V', F')$ and the functional dependency table

In classical Boolean networks, the states of nodes at time $t + 1$ depend on the states of nodes at time t, so that all nodes progress synchronously. Usually, in classical Boolean networks, the dynamics of the states of nodes are evolving according to the model structure and the scheme with an initial state. Hence, if the size of node n is fixed, the state space is finite (2^n). Therefore, given a particular set of nodes with the corresponding Boolean function, the trajectory or the state transition can be calculated. Consequently, a state will be eventually recur in the Boolean network. Since classical Boolean networks are completely deterministic, the dynamics are deterministic and a trajectory must reach a repeating state cycle. This means that the system of the network ultimately transits into an attractor. If an attractor consists of only a single state, it is called a point attractor or a steady state. If an attractor consists of two or more states, it is called a cycle attractor or a state cycle. The set of states that flow towards the same attractor state is called the basin of the attractor [38]. The effects of feedback for Boolean networks have been discussed in [31]. An example of a Boolean network with $n = 3$ is shown in Fig. 20.2.

Suppose there is a Boolean network $G(V, F)$ with n nodes $v_1, v_2, \ldots, v_n$ and initial joint probability distributions $D(v)$, $v \in \{0, 1\}^n$. The joint probability of node in the next time step would be

$$Pr\{f_1(v)=i_1, f_2(v)=i_2, \ldots, f_n(v) = i_n\} = \sum_{v\in\{0,1\}^n : f_j(v)=i_j, j=1,\ldots,n} D(v) \quad (20.1)$$

abc (t)	abc (t+1)
000	001
001	001
010	101
011	101
100	000
101	010
110	100
111	110

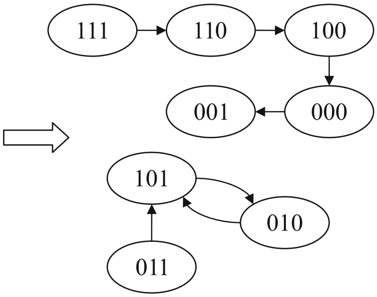


Fig. 20.2 The gene expression in time t and time t+1 is showed in the *left*, the transition space are showed in the *right*. There is one point attractor (001) with four states flowing into it, (111), (110), (100) and (001). There is also one cycle attractor of period two (101↔010), with one states flowing into it, (011)

The computing procedure could be iterative and the joint probability distribution in time step t could be described as $D^t(v) = \Psi(D^{t-1}(v))$ where the mapping Ψ is denoted by (1) and $D^{t-1}(v)$ is the joint probability distribution at time $t-1$. Therefore, the joint probability distribution at any time step t would be $D^t(v) = \Psi^t(D^0(v))$ where $D^0(v)$ is the initial joint probability distribution.

If we try to consider all possible networks, there will be 2^{2^k} possible functions for each node. In addition, each node has $n!/(n-k)!$ possible ordered combinations for k different links. Therefore, for each target gene, there are $2^{2^k} \cdot n!/(n-k)!$ possible input combinations to constitute a network. Hence, the number of possible networks with n nodes and k input links is the following [7]

$$(\frac{2^{2^k} n!}{(n-k)!})^n.$$

20.2.2 Reconstruction of Genetic Boolean Networks

Here we will discuss the network reconstruction problem with a Boolean network model. Let (I_j, O_j) be the pair of expression profiles for $\{v_1, \ldots, v_n\}$, where I_j is the expression at time t and O_j corresponds to the expression at time $t+1$. The network reconstruction problem is to reconstruct the classical Boolean network from a series of pair examples, $EX = \{(I_1, O_1), (I_2, O_2), \ldots, (I_m, O_m)\}$.

There are a variety of algorithms proposed for reconstructing the structure of a genetic regulatory network from expression data of genes under the model of classical Boolean networks [2, 19]. In this subsection, we will discuss one reconstruction algorithm called REVEAL proposed in [21]. First, we only consider Boolean networks in which the indegree of each node is bounded by a constant K, because it has been proven that the number of profiles required grows exponentially if k is not bounded [1]. For simplicity, we only show algorithms for the case of $K = 2$. However, the algorithm can be intuitively generalized to any K.

We start by computing the Shannon entropy that measures the systematic mutual information of the Boolean network state transition table in the algorithm [25]. The Shannon entropy is defined in term of the probability of an event P_i by

$$H = -\sum P_i \log P_i.$$

For a pair of binary elements (v_i, v_j), the individual and combined Shannon entropies are defined as

$$\begin{aligned} H(v_i) &= -\sum_{r=0,1} P(v_i = r)\log P(v_i = r), \\ H(v_j) &= -\sum_{s=0,1} P(v_j = s)\log P(v_j = s), \\ H(v_i, v_j) &= -\sum_{r,s=0,1} P(v_i = r, v_j = s)\log P(v_i = r, v_j = s). \end{aligned}$$

One example of a binary system for explaining the calculation of Shannon entropy is demonstrated in Fig. 20.3.

The conditional entropy $H(v_i|v_j)$ corresponds to the information contained in v_i but not shared with v_j. It can be shown that $H(v_i, v_j) = H(v_j) + H(v_i|v_j)$. If $H(v_i, v_j) = H(v_j)$, i.e. $H(v_i|v_j) = 0$, then all information contained in v_i is shared with v_j and we would think v_j can determine the expression of v_i completely. Next, we list and demonstrate the procedure of algorithm, REVEAL, for the problem of network reconstruction.

- Step 1: Calculation of entropies from input
  We calculate the entropy of every input node $\{v_i\}, i = 1, \ldots, n$. Since the indegree of each node is bounded by $K = 2$, we also need to calculate the entropies of each pair of input node $\{v_i, v_j\}, i, j = 1, \ldots, n$.
- Step 2: Identification of $k = 1$ links
  For each node $v_i' \in V, i = 1, \ldots, n$, we calculate the entropies of all single input-output pairs $H(v_i', v_j), i, j = 1, \ldots, n$. If $H(v_i', v_j) = H(v_j)$, then v_j

v_i	0	1	1	0	0	0	1	1	0	1
v_j	0	0	1	0	1	0	1	1	0	0

v_i/v_j	0	1	
0	0.4	0.1	0.5
1	0.2	0.3	0.5
	0.6	0.4	

$$\begin{aligned} H(v_i) &= -0.5\log(0.5) - 0.5\log(0.5) = 1, \\ H(v_j) &= -0.6\log(0.6) - 0.4\log(0.4) = 0.97, \\ H(v_i, v_j) &= -0.4\log(0.4) - 0.1\log(0.1) \\ &\quad -0.2\log(0.2) - 0.3\log(0.3) = 1.85. \end{aligned}$$

Fig. 20.3 Calculation of Shannon entropy for single element and pair elements. Probability is calculated by the frequency the state of on or off

completely determines v_i'. If there is no single input v_j such that $H(v_i', v_j) = H(v_j)$, execute Step 3, otherwise output v_j and constitute the rule between v_i' and v_j.

- Step 3: Identification of $k = 2$ links
  For each node $v_i' \in V, i = 1, \ldots, n$, we calculate the entropies of all pair inputs with one output $H(v_i', v_j, v_l), i, j, l = 1, \ldots, n$. If $H(v_i', v_j, v_l) = H(v_j, v_l)$, then the pair input (v_j, v_l) exactly determines v_i'. Then we constitute the rule between v_i' and v_j, v_l.

The advantage of this algorithm is its low time and memory complexity. We consider the example by the input/output pairs data as shown in Fig. 20.1. It is easy to reconstruct the classical Boolean network from the data $\{(I_1, O_1), (I_2, O_2), \ldots, (I_{16}, O_{16})\}$ by the algorithm REVEAL. We list the step-by-step demonstration in Fig. 20.4 from the network example of Fig. 20.1.

20.2.3 Analysis of the Computational Complexity and Required Number of Experiments

For the network reconstruction problem, we assess the time complexity of the REVEAL algorithm. If a node is controlled with exactly k input variables, there are $\binom{n}{k}$ possible combinations of input nodes. For the calculation of input entropies with indegree that is bounded by K, there are $\binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{K}$ input entropies that need to be evaluated. This constitute the computational complexity of $O(n^K)$. Moreover, for each node, there are $\binom{n}{k}$ entropies to be calculated in every step of the identification of k links. In total, there are K steps of identification because the indegree is bounded by K. Consequently, for each node, $\binom{n}{1} + \binom{n}{2} + \cdots + \binom{n}{K}$ entropies are evaluated in the step of identification with $k = 1, 2, \ldots, K$. Therefore, $O(n^{K+1})$ entropies are evaluated in total and the REVEAL algorithm works in polynomial time for fixed K. Besides, it has been shown that the $O(\log n)$ transition (INPUT/OUTPUT) pairs are necessary and sufficient for the network reconstruction with high probability if the maximum indegree of Boolean networks is bounded [2].

20.3 Probabilistic Boolean Networks

In the previous section, we have introduced the definition and properties of classical Boolean networks. However, the structure of classical Boolean networks has been criticized for its deterministic formality. If the state of every node is obtained at any one time step, the states of all nodes at next time step are determined and fixed in a classical Boolean network model. However, there may be scenarios in which a set of different Boolean functions is required to describe a transition between a set of variables. In this section, we are going to introduce the basic definition and notation for probabilistic Boolean networks which allow more than one possible Boolean function for each node and extend the network structure to a probabilistic setting.

Step 1. Input entropies

$H(v_1)$=1	$H(v_2)$=1	$H(v_3)$=1	$H(v_4)$=1
$H(v_1,v_2)$=2	$H(v_1,v_3)$=2	$H(v_1,v_4)$=2	$H(v_2,v_3)$=2
$H(v_2,v_4)$=2	$H(v_3,v_4)$=2		

Step 2. Identification of k=1 links

$H(v'_1,v_1)$=2	$H(v'_1,v_2)$=2	$H(v'_1,v_3)$=1	$H(v'_1,v_4)$=2
$H(v'_2,v_1)$=1.5	$H(v'_2,v_2)$=1.25	$H(v'_2,v_3)$=1.5	$H(v'_2,v_4)$=1.25
$H(v'_3,v_1)$=2	$H(v'_3,v_2)$=2	$H(v'_3,v_3)$=2	$H(v'_3,v_4)$=1
$H(v'_4,v_1)$=2	$H(v'_4,v_2)$=1	$H(v'_4,v_3)$=2	$H(v'_4,v_4)$=2

Rule table for v_1, v_3 and v_4

Input	Output	Input	Output	Input	Output
v_3	v_1	v_4	v_3	v_2	v_4
0	0	0	1	0	0
1	1	1	0	1	1

Step 3. Identification of k=2 links

$H(v'_2,v_1,v_2)$=2.5	$H(v'_2, v_1,v_3)$=2	$H(v'_2, v_1,v_4)$=2.5
$H(v'_2,v_2,v_3)$=2.5	$H(v'_2, v_2,v_4)$=2.5	$H(v'_2, v_3,v_4)$=2.5

Rule table for v_2

Input		Output
v_1	v_3	v_2
0	0	0
0	1	0
1	0	0
1	1	1

Fig. 20.4 The step-by-step demonstration of the algorithm REVEAL for network example showed in Fig. 20.1

20.3.1 Definition and Notation

Probabilistic Boolean networks were proposed in [26] as a generalization of the classical Boolean networks with more flexibility due to its non-deterministic structure. In a probabilistic Boolean network, every node v_i is assigned by a set $F_i = \{f_1^{(i)}, \dots f_{l(i)}^{(i)}\}$, where each $f_j^{(i)} : \{0,1\}^n \to \{0,1\}$ is a possible Boolean function determining the value of gene v_i. Clearly, a probabilistic Boolean network becomes

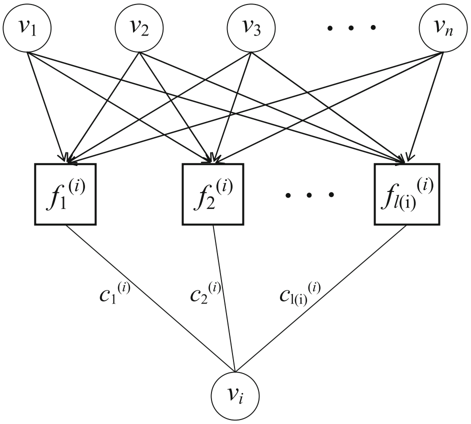


Fig. 20.5 The basic building block for the expression of gene v_i in a probabilistic Boolean model

a classical Boolean network if there is only one possible Boolean function for every node v_i, that is, the value of $l_{(i)}$ is 1, for all $i = 1, \ldots, n$.

We will also refer to the function $f_j^{(i)}$ as a predictor which is one of the possible Boolean function assigned to the expression of gene v_i. For every node v_i in V, one of the predictors in F_i would be selected randomly by a predefined probability distribution at any given time step. Therefore, at a given instant of time, a realization of a probabilistic Boolean network is determined by a vector of Boolean functions. We illustrate the basic building block for the expression of gene v_i of a probabilistic Boolean network in Fig. 20.5.

Suppose in total there are N different realizations in a probabilistic Boolean network, the N vector functions, $f_1, f_2, \ldots, f_N$ are defined as $f_m = (f_{m_1}^{(1)}, f_{m_2}^{(2)}, \ldots, f_{m_n}^{(n)})$, where $1 \leq m_i \leq l_{(i)}$ and $f_{m_i}^{(i)} \in F_i$ $(i = 1, 2, \ldots, n)$, for $m = 1, 2, \ldots, N$. Each vector function $f_m : \{0,1\}^n \to \{0,1\}^n$ represents a possible realization of the entire probabilistic Boolean networks. Hence, if the values of all genes $(v_1, v_2, \ldots, v_n)$ is known at time t and the realization f_m is chosen, $f_m(v_1, v_2, \ldots, v_n) = (v'_1, v'_2, \ldots, v'_n)$ gives us the state of the genes at time $t + 1$.

If the predictor for each gene is chosen independently, that is,

$$Pr\{f^{(i)} = f_{m_i}^{(i)}, f^{(j)} = f_{m_j}^{(j)}\} = Pr\{f^{(i)} = f_{m_i}^{(i)}\}Pr\{f^{(j)} = f_{m_j}^{(j)}\},$$

for all i, j, m_i, m_j with $1 \leq m_i \leq l_{(i)}$, $1 \leq m_j \leq l_{(j)}$, then the probabilistic Boolean network is said to be pairwise independent. Under the assumption of independence of the random variables $f^{(1)}, f^{(2)}, \ldots, f^{(n)}$, the number of possible probabilistic Boolean network realizations is $N = \prod_{i=1}^{n} l(i)$ [26].

Although the domain of each predictor function $f_j^{(i)}$ is $\{0,1\}^n$, there should be many fictitious variables that are not needed in the function. A variable v_i is described as fictitious for a function f, if the state of v_i would not affect the output of function f, that is,

$$f(v_1, \ldots, v_{i-1}, 0, v_{i+1}, \ldots, v_n) = f(v_1, \ldots, v_{i-1}, 1, v_{i+1}, \ldots, v_n) \quad (20.2)$$

for all $v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n$. Consequently, there are only a few input genes that actually regulate gene x_i at any given time. Let f be a random vector, representing the realization of a probabilistic Boolean network, then $f = (f^{(1)}, f^{(2)}, \ldots, f^{(n)})$, where $f^{(i)} \in F_i$ for all $i = 1, 2, \ldots, n$. Hence, for a node v_i, the probability that $f_j^{(i)}$ is selected as the predictor $(1 \le j \le l(i))$ is

$$c_j^{(i)} = P(f^{(i)} = f_j^{(i)}) = \sum_{m: f_{m_i}^{(i)} = f_j^{(i)}} Pr\{f = f_m\} \quad (20.3)$$

If we define the $N \times n$ matrix M as

$$M = \begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ 1 & 1 & \cdots & 1 & 2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & l(n) \\ 1 & 1 & \cdots & 2 & 1 \\ 1 & 1 & \cdots & 2 & 2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 2 & l(n) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ l(1) & l(2) & \cdots & l(n-1) & l(n) \end{pmatrix}.$$

each one corresponding to a possible network configuration, then the probability of network i being selected is

$$P_i = Pr\{\text{Network i is selected}\} = \prod_{j=1}^{n} c_{M_{ij}}^{(j)}, \quad (20.4)$$

where M_{ij} is the ijth entry in matrix M.

Next, we establish a $2^n \times 2^n$ state transition matrix A by

$$\begin{aligned} A(v, v') &= Pr\{(v_1, \ldots, v_n) \to (v'_1, \ldots, v'_n)\} \\ &= \sum_{i: f_{K_{i1}}^{(1)}(v_1, \ldots, v_n) = v'_1, \ldots, f_{K_{in}}^{(n)}(v_1, \ldots, v_n) = v'_n} P_i \end{aligned} \quad (20.5)$$

It was shown that the state transition matrix A is a Markov matrix and the probabilistic Boolean network is a homogeneous Markov process [26].

Let us illustrate the above construction with an example. We consider a probabilistic Boolean network consisting of three genes $V = \{v_1, v_2, v_3\}$ and the function sets $F = \{F_1, F_2, F_3\}$ with $F_1 = \{f_1^{(1)}\}$, $F_2 = \{f_1^{(2)}, f_2^{(2)}\}$, $F_3 = \{f_1^{(3)}, f_2^{(3)}, f_3^{(3)}\}$. The rule of each function is given by the following truth table.

By assuming the independence of the probabilistic Boolean network, there are six possible realizations in this example and the matrix M becomes

$$M = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 1 & 3 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \end{pmatrix}.$$

The network i represented by the ith row of M is selected meaning that the predictors $(f_{M_{i1}}^{(1)}, f_{M_{i2}}^{(2)}, f_{M_{i3}}^{(3)})$ will be used.

Let P_i be the probability that network i is selected. In this example, the state transition matrix A is given by

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & P_4 + P_5 + P_6 & 0 & P_1 + P_2 + P_3 \\ 0 & 0 & P_2 + P_3 + P_5 + P_6 & P_1 + P_4 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & P_2 + P_5 & P_1 + P_3 + P_4 + P_6 & 0 & 0 & 0 & 0 \\ P_5 + P_6 & P_4 & P_2 + P_3 & P_1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

20.3.2 Inference of Probabilistic Boolean Networks

For a given gene, there is a set of predictors that could be selected. One approach to estimate the probability is the method of coefficient of determination (COD) which was used in an optimal nonlinear filter design [5, 18]. Let $\boldsymbol{v}_1$ be a target gene that we wish to predict by a set of predictors function $f_1^{(i)}, f_2^{(i)}, \ldots, f_{l(i)}^{(i)}$. For each predictor $f_j^{(i)}$, one can use the COD to find a set of genes $\boldsymbol{v}_j^{(i)}$ such that $f_j^{(i)}(\boldsymbol{v}_j^{(i)})$ are the optimal predictors.

Specifically, the CODs for v_i related to the predictor $f_j^{(i)}(\boldsymbol{v}_j^{(i)})$ for each j is defined as

$$\theta_j^i = \frac{\epsilon_i - \epsilon(v_i, f_j^{(i)}(v_j^{(i)}))}{\epsilon_i},$$

where ϵ_i is the error of the best estimate of v_i without any conditional variables and $\epsilon(v_i, f_j^{(i)}(v_j^{(i)}))$ is the error measure of v_i given the predictor $f_j^{(i)}$. It is clear that the value of COD is between 0 and 1. The large value of COD indicate higher evidence that the corresponding predictor with its input genes are plausible.

Let us now assume predictors $f_1^{(i)}, f_2^{(i)}, \ldots, f_{l(i)}^{(i)}$ with a class of gene sets $\boldsymbol{v}_1^{(i)}$, $\boldsymbol{v}_2^{(i)}, \ldots, \boldsymbol{v}_{l(i)}^{(i)}$ are selected with the highest CODs. Then, for a given gene v_i, the probability that predictor $f_j^{(i)}$ is selected is estimated by

$$C_j^{(i)} = \frac{\theta_j^i}{\sum_{j=1}^{l(i)} \theta_j^i}.$$

The number of predictors, $l(i)$, is a parameter selected by the user based on the amount of training data available and existing biological information.

20.3.3 Influences Between Pairs of Genes in Probabilistic Boolean Networks

In the probabilistic Boolean network model, every gene is influenced by a set of Boolean functions with several input genes. However, the contribution of every input gene is not always the same. Hence, it is important to distinguish genes that have major impacts on the predictor from those that have minor impacts.

Since every gene is controlled by a set of Boolean functions, we first consider the influence of variables on a Boolean function. One method is to quantify the influence of a variable on a Boolean function, as proposed in [26,28]. The influence of the variable v_j on the function f is defined as

$$I_j(f) = E_D[\frac{\partial f}{\partial v_j}] = Pr\{f(v) \neq f(v^{(j)})\}$$

where $E_D[\,]$ is the expectation operator with respect to distribution $D(v)$ and $v^{(j)}$ is the same as v with jth component is toggled (from 0 to 1, or from 1 to 0). For a probabilistic Boolean network, F_i denotes the set of predictor for gene v_i with corresponding probabilities $c_1^{(i)}, \ldots, c_{l(i)}^{(i)}$. Hence, the influence of gene v_m on gene v_i can be defined as

$$I_m(v_i) = \sum_{j=1}^{l(i)} I_m(f_j^{(i)}) c_j^{(i)},$$

Table 20.1 One example of probabilistic Boolean network model

$x_1x_2x_3$	$f_1^{(1)}$	$f_1^{(2)}$	$f_2^{(2)}$	$f_1^{(3)}$	$f_2^{(3)}$	$f_3^{(3)}$
000	1	0	0	1	1	1
001	1	1	0	1	1	1
010	0	1	1	1	0	0
011	0	0	0	0	0	0
100	1	0	0	1	1	1
101	1	1	1	1	1	1
110	0	1	1	1	0	1
111	0	1	0	1	0	0

where $I_m(f_j^{(i)})$ is the influence of variable x_m on the predictor $f_j^{(i)}$. The influence matrix Γ with elements of $\Gamma_{ij} = I_i(v_j)$ collects the information of influence between every pair of genes.

We consider the probabilistic Boolean network shown in Table 20.1 with $c_1^{(1)}=1$, $c_1^{(2)} = 0.4$, $c_2^{(2)} = 0.6$, $c_1^{(3)} = 0.2$, $c_2^{(3)} = 0.4$, $c_3^{(3)} = 0.4$. We let the initial joint probability distribution D be an uniform distribution, that is, $D(v) = 1/8$ for all $v \in [0,1]^3$. Suppose we would like to compute the influence of variable v_1 on variable v_2, we need to calculate the influence of v_1 on the predictor $f_1^{(2)}$ and $f_2^{(2)}$ by

$$I_1(f_1^{(2)}) = E_D[\frac{\partial f_1^{(2)}(v)}{\partial v_1}] = 0.25,$$

$$I_1(f_2^{(2)}) = E_D[\frac{\partial f_2^{(2)}(v)}{\partial v_1}] = 0.25.$$

Hence, the influence of variable v_1 on variable v_2 would be

$$I_1(v_2) = I_1(f_1^{(2)}) \cdot c_1^{(2)} + I_1(f_2^{(2)}) \cdot c_2^{(2)} = 0.4 \cdot 0.25 + 0.6 \cdot 0.25 = 0.25.$$

By computing every pair of gene similar to the process above, we can obtain the influence matrix

$$\Gamma = \begin{pmatrix} 0 & 0.25 & 0.15 \\ 1 & 0.75 & 0.75 \\ 0 & 0.75 & 0.15 \end{pmatrix}.$$

20.4 Directed Acyclic Boolean Networks

In the previous section, we introduced the classical Boolean network model and the probabilistic Boolean network model in order to analyze the expression profiles of genes with time courses. However, in some experiments, the expressions

Table 20.2 The table of states for directed acyclic Boolean network shown in Fig. 20.6

Case	1	2	3	4	5	6	7
v_1	0	1	1	1	1	1	1
v_2	0	0	0	0	1	1	1
v_3	0	0	1	1	0	1	1
v_4	0	0	0	0	1	1	1
v_5	0	0	0	0	0	0	1

of genes only can be observed at a specific time. Therefore, in this section, we will discuss a directed acyclic Boolean network model for handling the static and dynamic expression profiles of genes [20]. A directed acyclic Boolean network is uniquely determined by the state space of its elements: all possible on-off states that are compatible with the network structure. Our goal is to reconstruct directed acyclic Boolean networks from possibly noisy array data.

20.4.1 The Structure of Directed Acyclic Boolean Networks

Firstly, we will introduce the structure of directed acyclic Boolean networks. Suppose there are m elements, $v_1, v_2, \ldots, v_m$ in a Boolean network. For any two elements v_i and v_j, we have two kinds of pairwise relationships: prerequisite and similarity. We say that v_i is prerequisite for v_j if the on-status of v_i is necessary for the on-status of v_j and this relationship is denoted by $v_i \prec v_j$. That is, if we know the status of v_j is 1 and $v_i \prec v_j$, then the status of v_i must be 1. Therefore for any pair of elements (v_i, v_j) with a prerequisite relation, there are a total of four possible relationships: $v_i \prec v_j$, $v_i \prec \bar{v}_j$, $\bar{v}_i \prec v_j$ and $\bar{v}_i \prec \bar{v}_j$. The prerequisite relationship is transitive, thus if $v_i \prec v_j$ and $v_j \prec v_k$, then we have $v_i \prec v_k$. For any pair of elements (v_i, v_j) which have prerequisite relationship $v_i \prec v_j$, we say they are covering pair if there are no other element v_k such that $v_i \prec v_k$ and $v_k \prec v_j$.

The other types of relationships between pairs of elements is similarity and negative similarity. We say that v_i and v_j are similar if the status of two elements is consistent. That is, the status of these two elements is on and off simultaneously, and this is denoted by $v_i \sim v_j$. There is another relationship of negative similarity and there are two possible relationships: $v_i \sim v_j$ and $v_i \sim \bar{v}_j$.

In the diagram, if v_i is prerequisite to v_j, we draw a directed arrow from the vertex v_i to v_j, and if v_i is similar to v_j, we use an undirected line to connect v_i and v_j. For the purpose of making the prerequisite relationships more clear in the graph, we only represented all partial orders by arrows between covering pairs.

We will illustrate the above construction by an example with Fig. 20.6 which has five elements with one similarity and four prerequisite relationship. For five Boolean elements, there are totally $2^5 = 32$ possibilities in the state space and only seven states are compatible with the diagram.

Fig. 20.6 A diagram of directed acyclic Boolean network with the corresponding table of states

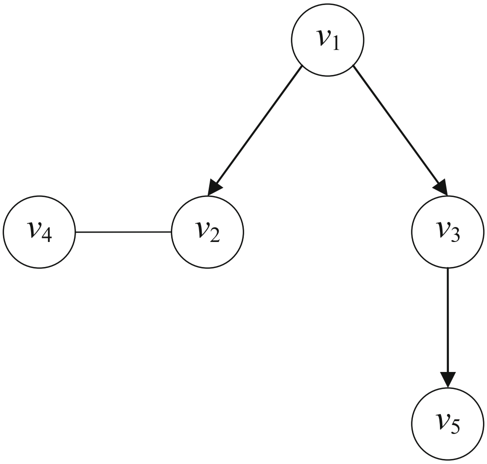


Table 20.3 Count patterns for the basic six relationships assuming exhaustive sampling and no measurement error

$v_i \sim v_j$			$v_i \prec \bar{v}_j$			$\bar{v}_i \prec v_j$		
v_i/v_j	0	1	v_i/v_j	0	1	v_i/v_j	0	1
0	+	0	0	0	+	0	+	+
1	0	+	1	+	+	1	+	0
$v_i \sim \bar{v}_j$			$v_i \prec v_j$			$\bar{v}_i \prec \bar{v}_j$		
v_i/v_j	0	1	v_i/v_j	0	1	v_i/v_j	0	1
0	0	+	0	+	0	0	+	+
1	+	0	1	+	+	1	0	+

The seven states that are compatible with Fig. 20.6 are enumerated in Table 20.2. Suppose we generate n samples from the directed acyclic Boolean network in Fig. 20.6. That is, we sample with replacement from Table 20.2 and arrange the data in a matrix (y_{ij}), where $i = 1, \ldots, n$, $j = 1, \ldots, 5$. We can identify the relationship of each pair of elements as prerequisite or as similar relationships from the corresponding two columns of data matrix (y_{ij}), which is the transposition of Table 20.2. Then, the directed acyclic Boolean network would be reconstructed by integrating the pair relationships together. For each pair of elements, we count the frequencies in four cells of (v_i, v_j) for $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$ from the samples and arrange them in a 2×2 table. We mark a cell '$+'$ if the frequency count is positive and mark it '0' otherwise. Then, we check the table with those six pairwise relationships listed in Table 20.3. For example, if we consider the paired elements v_1 and v_2, the frequency counts of (v_1, v_2) for $(0,0)$, $(1,0)$ and $(1,1)$ are positive and there is no incident for $(0,1)$, therefore, the relationship between v_1 and v_2 would be $v_1 \prec v_2$.

20.4.2 Identification Algorithm with Noisy Array

In Sect. 20.4.1, we discussed an identification method for data without noise. In this subsection we will consider the situation of noisy array data. We assume that every element in the entry of (y_{ij}), $j = 1, 2, \ldots, m$ switches to its reverse status with a misclassification probability p independently; that is

$$x_{ij} = \begin{cases} y_{ij} & \text{with probability } 1 - p; \\ 1 - y_{ij} & \text{with probability } p. \end{cases} \tag{20.6}$$

Thus, the observed array (x_{ij}) contains misclassification error. Our goal is to reconstruct directed acyclic Boolean networks from noisy array of binary data (x_{ij}).

In the first step, we investigate every pair of elements for possible relationships. Next, we use the probabilistic model of equation (20.6) to estimate misclassification probability p. We treat the data in the 2 × 2 table as a multinomial distribution with four cells whose probabilities are $q_{00}, q_{01}, q_{10}, q_{11}$, respectively, where $q_{00} + q_{01} + q_{10} + q_{11} = 1$.

The observed data $n_{00}, n_{01}, n_{10}, n_{11}$ are generated from the multinomial distribution with probability $r_{00}, r_{01}, r_{10}, r_{11}$, where $r_{00} + r_{01} + r_{10} + r_{11} = 1$. The relationship between q_{ij} and r_{ij} is displayed in Table 20.5 and explained below.

Because of the misclassification error, a portion of samples of m_{00} may change to the other three cells. We use the notations of $m_{00,00}, m_{00,01}, m_{00,10}, m_{00,11}$ to represent the counts of four cells changed from m_{00}. Analogous notations are defined for m_{01}, m_{10} and m_{11}. Consequently, their generating probabilities $(q_{00}, q_{01}, q_{10}, q_{11})$ are calculated as follows: $q_{ij,kl} = p^{|i-k|+|j-l|}(1-p)^{2-|i-k|-|j-l|}q_{ij}$. Here, we adopt the notation $q_{ij,kl}$ analogous to $m_{ij,kl}$. The above parameters and splits are shown in Tables 20.4 and 20.5. By these two table, it is easy to find that the correspondence between two sets of counts and probabilities is the following:

$$\begin{cases} n_{kl} = \sum\limits_{i,j=0,1} m_{ij,kl}, \\ r_{kl} = \sum\limits_{i,j=0,1} q_{ij,kl}; \end{cases}$$

Table 20.4 Splitting counts caused by misclassification error

(v_i, v_j)	Observed				
Actual	00	01	10	11	
00	$m_{00,00}$	$m_{00,01}$	$m_{00,10}$	$m_{00,11}$	m_{00}
01	$m_{01,00}$	$m_{01,01}$	$m_{01,10}$	$m_{01,11}$	m_{01}
10	$m_{10,00}$	$m_{10,01}$	$m_{10,10}$	$m_{10,11}$	m_{10}
11	$m_{11,00}$	$m_{11,01}$	$m_{11,10}$	$m_{11,11}$	m_{11}
	n_{00}	n_{01}	n_{10}	n_{11}	n

Table 20.5 Splitting probabilities caused by the misclassification error

(v_i, v_j)	Observed				
Actual	00	01	10	11	
00	$q_{00,00} = (1-p)^2 q_{00}$	$q_{00,01} = p(1-p)q_{00}$	$q_{00,10} = p(1-p)q_{00}$	$q_{00,11} = p^2 q_{00}$	q_{00}
01	$q_{01,00} = p(1-p)q_{01}$	$q_{01,01} = (1-p)^2 q_{01}$	$q_{01,10} = p^2 q_{01}$	$q_{01,11} = p(1-p)q_{01}$	q_{01}
10	$q_{10,00} = p(1-p)q_{10}$	$q_{10,01} = p^2 q_{10}$	$q_{10,10} = (1-p)^2 q_{10}$	$q_{10,11} = p(1-p)q_{10}$	q_{10}
11	$q_{11,00} = p^2 q_{11}$	$q_{11,01} = p(1-p)q_{11}$	$q_{11,10} = p(1-p)q_{11}$	$q_{11,11} = (1-p)^2 q_{11}$	q_{11}
	r_{00}	r_{01}	r_{10}	r_{11}	1

and

$$\begin{cases} m_{ij} = \sum\limits_{k,l=0,1} m_{ij,kl}, \\ q_{ij} = \sum\limits_{k,l=0,1} q_{ij,kl}. \end{cases} \tag{20.7}$$

For the complete data $\{m_{ij,kl}\}$, the log-likelihood is given by

$$L = \sum_{i,j,k,l=0,1} m_{ij,kl} \log q_{ij,kl}, \tag{20.8}$$

where $q_{ij,kl}$ are those splitting probabilities. Since the complete data $\{m_{ij,kl}\}$ are not observable, we use the E-M algorithm to maximize the log-likelihood. In the E-step, the splitting counts of complete data $\{m_{ij,kl}\}$ are evaluated by the conditional expectations using the current values of the parameters by the following formula

$$E_{p,q_{00},q_{01},q_{10},q_{11}}(m_{ij,kl}|n_{kl}) = \frac{n_{kl} q_{ij,kl}}{\sum\limits_{i'j'=0,1} q_{i'j',kl}}, \tag{20.9}$$

where $i, j, k, l = 0, 1$. One or two probabilities of $q_{00}, q_{01}, q_{10}, q_{11}$ are zero in those different hypotheses specified in Table 20.6. In the M-step, we maximize the conditional expectation of the log-likelihood for the complete data to obtain the maximum likelihood estimates (MLEs) of the parameters. According to the MLEs, we can compute the p-score or s-score for every pair of elements, which are obtained by the estimate for the misclassification probability under prerequisite or similar relationship.

For the first step, we would like to determine the most probable relationships between elements and select candidate pairs of genes for the watch list. Next, we reconstruct a directed acyclic Boolean network by integrating the relationship of those genes selected.

For a pair of genes v_i and v_j, we define the p-scores $p_{v_i \prec \bar{v}_j}$, $p_{v_i \prec v_j}$, $p_{\bar{v}_i \prec \bar{v}_j}$, $p_{\bar{v}_i \prec v_j}$ are, respectively, the maximum likelihood estimates of p under the triangular model: $q_{00} = 0, q_{01} = 0, q_{10} = 0, q_{11} = 0$. The s-scores $s_{v_i \sim v_j}$ and $s_{v_i \sim \bar{v}_j}$ are the maximum likelihood estimates of p under the diagonal model: $q_{01} = q_{10} = 0$ and $q_{00} = q_{11} = 0$, respectively.

Table 20.6 The six basic relationships and their corresponding probabilistic hypotheses and scores

Relation	Hypothesis	Scores
$v_i \prec \bar{v}_j$	$q_{00} = 0$	$p_{v_i \prec \bar{v}_j}$
$v_i \prec v_j$	$q_{01} = 0$	$p_{v_i \prec v_j}$
$\bar{v}_i \prec v_j$	$q_{10} = 0$	$p_{\bar{v}_i \prec v_j}$
$\bar{v}_i \prec \bar{v}_j$	$q_{11} = 0$	$p_{\bar{v}_i \prec \bar{v}_j}$
$v_i \sim \bar{v}_j$	$q_{01} = q_{10} = 0$	$s_{v_i \sim \bar{v}_j}$
$v_i \sim v_j$	$q_{00} = q_{11} = 0$	$s_{v_i \sim v_j}$

According to the E-M algorithm described above, we can evaluate the s-score and p-score for every pair of elements. We use the MLE $\hat{p}$ to measure how well each hypothesis fits: the smaller the score, the more evidence that the corresponding hypothesis could be true.

For each pair of elements, we find the diagonal model which have the smaller s-score and the triangular model which have the smallest p-score. Then we evaluate their BIC values by

$$BIC = -\log likelihood + \frac{d \log n}{2},$$

where d is the number of parameters for one possible relationship. We treat the model with the smaller BIC value as the most probable relationship for the pair elements and the s-p-score is defined as the corresponding score under the model. Next, for every pair of elements, we rank its s-p-score in the ascending order. The smaller the s-p-score is, the more likely the relationship could be true.

If the samples are generated from a directed acyclic Boolean network, s-p-scores are quite useful for the discovery of pairwise relationships. Here we could consider the *maximum compatibility criterion*: to choose the maximum threshold value so that the selected relationships contain no conflicts [20]. We collect those relationships whose s-p-scores are smaller than a threshold. Known biological results could be helpful for the determination of a threshold. For example, if we know the relationship $v_1 \prec v_3$ is true, then the s-p-scores smaller than $p_{v_1 \prec v_3}$ should be in our watch list. As more relationships are included in the watch list, the more likely we are to observe incompatible ones. In general, we can choose the threshold which allows the maximum number of relationships with no conflicting relationships.

We now evaluate the computational complexity of statistical reconstruction method of SPAN described above. The key procedure is the computation of s-p-score for every pair of elements. If the number of elements is m, their are totally $\binom{m}{2}$ pairs of elements and the complexity for the computation of MLE is $O(m^2)$. We can rank the s-p-score of every pair elements in the order of $O(m^2 \log m)$. Thus, in this statistical reconstruction algorithm, the time complexity is $O(m^2 \log m)$ and the memory complexity is $O(m^2)$ as described in [20].

20.5 Conclusion

We have introduced a variety of models including classical Boolean networks, probabilistic Boolean networks and directed acyclic Boolean networks for dealing with genetic regulatory networks. These variants of Boolean networks can be used in the exploration of large genetic networks because of the simple structure of Boolean networks. Based on the reconstruction of Boolean networks, more flexible models, like Bayesian networks, can be applied to investigate more complex problems.

There are several advantages in estimating gene regulatory networks with Boolean networks. First of all, a variety of software packages have recently been developed for constructing Boolean networks. Matlab implementations of classical Boolean network toolbox and for probabilistic Boolean networks were developed in [24, 26]. Moreover, Li and Lu also provided an implementation for the s-p-scoring method in Matlab [20]. Other genetic regulatory network tools such as NetBuilder for simulating genetic Boolean network are also available [35]. Second, recent research indicates that various complex biological processes can be described by seemingly simplistic Boolean formalisms [33, 34]. The dynamic behaviors of living systems can be explained effectively by Boolean networks [9, 29]. Moreover, for large-scale gene regulatory networks, Kim et al. [17] have used Boolean network with chi-square test on the yeast cell cycle microarray gene expression data sets. Kauffman et al. [16] have used a random Boolean network to get possible interaction rules for transcriptional network models in yeast. Furthermore, the dynamic behaviors of cellular states are also represented by attractors in Boolean network in [9].

One characteristic of a Boolean network is that all the variables in the graph are binary. If the data we observed is continuous or quantized to have more than two levels, we need to discretize them. For microarray data, the ratios of expression level would be one possible approach of discretization. That is, we can treat the gene as on (active) if the log-ratio of its expression is larger than zero and off (inactive) otherwise. In general, biological background knowledge will be helpful for setting thresholds for discretizaion. On the other hand, if the samples are obtained from a time course, then we can consider the gene as on or off by detecting the gene is either increasing or decreasing with time.

For future developments on Boolean networks, we can consider more complicated structures such as Boolean networks with time delay. Furthermore, we can develop models of Boolean networks that have more flexible structures than these models proposed in literature. Since Boolean network models have been shown to be useful for reconstructing genetic network from real biological gene expression profiles, the evaluation of Boolean network models' effectiveness will be an important task in the future.

Acknowledgements The authors would like to express their gratitude to the English editing of Yang Wang and Arthur Tu. This work was partially supported by the National Science Council (NSC), National Center for Theoretical Sciences (NCTS) and Center of Mathematical Modeling and Scientific Computing (CMMSC) at the National Chiao Tung University in Taiwan.

References

1. Akutsu, T., Kuhara, S., Maruyama, O., & Miyano, S. (1998). Identification of gene regulatory networks by strategic gene disruptions and gene overexpression. In *Proceeding 9th ACM-SIAM symposium discrete algorithms* (pp. 695–702).
2. Akutsu, T., & Miyano, S. (1999). Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Pacific Symposium on Biocomputing*, *4*, 17–28.
3. Akutsu, T., Miyano, S., & Kuhara, S. (2000). Inferring qualitative relations genetic networks and metabolic pathways. *Bioinformatics*, *16*, 727–734.
4. Bornholdt, S. (2005). Less is more in modeling large genetic networks. *Science*, *310*(5747), 449–451.
5. Dougherty, E. R., Kim, S., & Chen, Y. (2000). Coefficient of determination in nonlinear signal processing. *Signal Processing*, *80*, 2219–2235.
6. Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, *7*, 601–620.
7. Harvey, I., & Bossomaier, T. (1997). Time out of joint: Attractors in asynchronous random Boolean network. In *Proceedings of the fourth European conference on artificial life* (pp. 67–75).
8. Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, *20*, 197–243.
9. Huang, S. (1999). Gene expression profiling, genetic networks and cellular states: An integrating concept for tumorigenesis and drug discovery. *Journal of Molecular Medicine*, *77*, 469–480.
10. Imoto, S., Goto, T., & Miyano, S. (2002). Estimation of genetic networks and functional structures between genes by using Bayesian network and nonparametric regression. *Pacific Symposium on Biocomputing*, *7*, 175–186.
11. Imoto, S., Higuchi, T., Goto, T., Tashiro, K., Kuhara, S., & Miyano, S. (2004). Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. *Journal of Bioinformatics and Computational Biology*, *2*, 77–98.
12. Jensen, F. V. (1996). *An introduction to Bayesian networks*. London: University College London Press.
13. Jensen, F. V. (2001). *Bayesian networks and decision graphs*. New York: Springer.
14. Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, *22*(3), 437–467.
15. Kauffman, S. A. (1993). *The origins of order: Self-organization and selection in evolution*. New York: Oxford University Press.
16. Kauffman, S. A., Peterson, C., Samuelsson, B., & Troein, C. (2003). Random Boolean network models and the yeast transcriptional network. *Biophysics*, *100*(25), 14796–14799.
17. Kim, H., Lee, J. K., & Park, T. (2007). Boolean networks using the chi-square test for inferring large-scale gene regulatory networks. *BMC Bioinformatics*, *8*, 37.
18. Kim, S., Dougherty, E. R., Chen, Y., Sivakumar, K., Meltzer, P., Trent, J. M., & Bittner, M. (2000). Multivariate measurement of gene expression relationships. *Genomics*, *67*, 201–209.
19. Laubenbacher, R., & Stigler, B. (2004). A computational algebra approach to the reverse engineering of gene regulatory networks. *Journal of Theoretical Biology*, *299*, 523–537.
20. Li, L. M., & Lu, H. H.-S. (2005). Explore biological pathways from noisy array data by directed acyclic Boolean networks. *Journal of Computational Biology*, *12*(2), 170–185.
21. Liang, S., Fuhrman, S., & Somogyi, R. (1998). REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing*, *3*, 18–29.
22. Moler, E. J., Radisky, D. C., & Mian, I. S. (2000). Integrating naive Bayes models and external knowledge to examine copper and iron homeostasis in *S. cerevisiae*. *Physiol Genomics*, *4*(2), 127–135.

23. Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. San Mateo: Morgan Kaufmann.
24. Schwarzer, C., & Teuscher C. (2003). *The software of teuscher's Lab: Matlab random Boolean network toolbox.* Swiss Federal Institute of Technology Lausanne (EPFL). URL http://www.teuscher.ch/rbntoolbox/
25. Shannon, C. E., & Weaver, W. (1963). *The Mathematical Theory of Communication.* University of Illinois Press. ISBN: 0252725484.
26. Shmulevich, I., Dougherty, E. R., Kim, S., & Zhang, W. (2002). Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, *18*(2), 261–274.
27. Shmulevich, I., Dougherty, E. R., & Zhang, W. (2002). From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proceeding of the IEEE*, *90*(11), 1778–1792.
28. Shmulevich, I., Dougherty, E. R., & Zhang, W. (2002). Gene perturbation and intervention in probabilistic Boolean networks. *Bioinformatics*, *18*(10), 1319–1331.
29. Shmulevich, I., Gluhovsky, I., Hashimoto, R. F., Dougherty, E. R., & Zhang, W. (2003). Steady-state analysis of genetic regulatory networks modelled by probabilistic Boolean networks. *Comparative and Functional Genomics*, *4*, 601–608.
30. Somogyi, R., & Sniegoski, C. A. (1996). Modeling the complexity of genetic networks: Understanding multigene and pleiotropic regulation. *Complexity*, *1*, 45–63.
31. Sontag, E., Veliz-Cuba, A., Laubenbacher, R., & Jarrah, A. S. (2008). The effect of negative feedback loops on the dynamics of Boolean networks. *Biophysical Journal*, *95*, 518–526.
32. Spirtes, P., Glymour, C., & Scheines, R. (2000). Causation, prediction and search. Cambridge, MA: MIT.
33. Szallasi, Z., & Liang, S. (1998). Modeling the normal and neoplastic cell cycle with 'realistic Boolean genetic networks': Their application for understanding carcinogenesis and assessing therapeutic strategies. *Pacific Symposium on Biocomputing*, *3*, 66–76.
34. Thomas, R., Thieffry, D., & Kaufman, M. (1995). Dynamical behaviour of biological regulatory networksXI. Biological role of feedback loops and practical use of the concept of the loop-characteristic state. *Bulletin of Mathematical Biology*, *57*(2), 247–276.
35. Wegner, K., Knabe, J., Robinson, M., Egri-Nagy, A., Schilstra, M., & Nehaniv, C. (2007). The NetBuilder'project: development of a tool for constructing, simulating, evolving, and analysing complex regulatory networks. *BMC Systems Biology*, *1*(Suppl 1):P72.
36. Wolfram, S. (1983). Statistical mechanics of cellular automata. *Reviews of Modern Physics*, *55*(3), 601–644.
37. Wolfram, S. (1984). Universality and complexity in cellular automata. *Physica 10D*, *10*(1), 1–35.
38. Wuensche, A. (1998). Genomic regulation modeled as a network with basins of attraction. *Pacific Symposium on Biocomputing*, *3*, 89–102.

Chapter 21
Protein Interaction Networks: Protein Domain Interaction and Protein Function Prediction

Yanjun Qi and William Stafford Noble

Abstract Most of a cell's functional processes involve interactions among proteins, and a key challenge in proteomics is to better understand these complex interaction graphs at a systems level. Because of their importance in development and disease, protein-protein interactions (PPIs) have been the subject of intense research in recent years. In addition, a greater understanding of PPIs can be achieved through the detailed investigation of the protein domain interactions which mediate PPIs. In this chapter, we describe recent efforts to predict interactions between proteins and between protein domains.

We also describe methods that attempt to use protein interaction data to infer protein function. Protein-protein interactions directly contribute to protein functions, and implications about functions can often be made via PPI studies. These inferences are based on the premise that the function of a protein may be discovered by studying its interaction with one or more proteins of known functions. The second part of this chapter reviews recent computational approaches to predict protein functions from PPI networks.

21.1 Introduction

In recent years, the human and other genome sequencing projects have generated vast amounts of data that identify thousands of new gene products whose functions and interrelationships are not yet known. The overall molecular architecture of all organisms is largely mediated both structurally and functionally through the coordination of protein-protein interactions (PPIs). In particular, the disruption of PPIs

Y. Qi
Machine Learning Department, NEC Labs America
e-mail: yanjun@nec-labs.com

W.S. Noble (✉)
Department of Genome Sciences, Department of Computer Science and Engineering,
University of Washington
e-mail: williamnoble@uw.edu

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_21,
© Springer-Verlag Berlin Heidelberg 2011

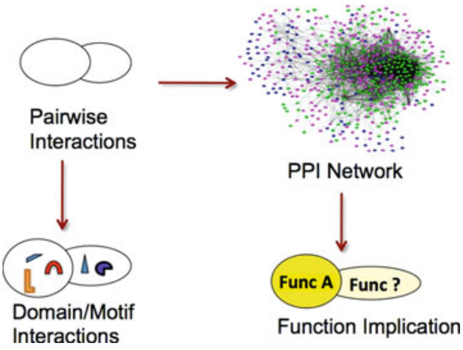


Fig. 21.1 The framework of contents in this chapter

may lead to the development of diseases. Thus, correctly identifying the interrelationship between proteins at the systems level is urgent and necessary, since such knowledge would lead to a better understanding of the functional properties that define the behaviors of most complex biological systems.

Experimental techniques [81] to detect PPIs or protein functions have their own limitations, and the resulting data sets are often noisy. Thus, additional approaches are needed to accelerate the recovery of complex protein-interaction systems. Given the vast amount of available biological evidence and the representational power of mathematical models, computational methods are gaining importance. In this chapter, we review three areas to which computational approaches contribute significantly (Fig. 21.1). We first introduce methods targeting protein-protein interaction predictions in Sect. 21.2. Then in Sect. 21.3 recent advances in identifying domain-domain interactions are presented. Finally, Sect. 21.4 reviews various ways to predict protein functions from PPI graphs.

21.2 Prediction of Protein-Protein Interactions

The term “protein-protein interactions” refers to the association of protein molecules with each other. The associations are interesting from multiple perspectives, including ascertainment of specific biological processes and pathways such as signal transduction pathways, as well as the systems-level studies of networks on the cellular or organism-wide scale. Because direct pairwise PPIs provide the basic building blocks to carry out the myriad of functions in a cell, comprehensively identifying these interactions is essential for understanding the molecular mechanisms underlying biological functions.

Experimental techniques for deciphering protein-protein interactions have been reviewed by [81]. In general, interactions among proteins can take on many forms (e.g., have an impact on functions of one another, or occur in a common pathway), and many proteins only operate in complexes and through physical contact with other proteins. These factors have prompted the development of various complementary experimental methods for detecting protein-protein interactions. Traditionally, PPIs have been studied individually through the use of genetic, biochemical and biophysical experimental techniques (also termed *small-scale* methods). The related experiments are generally time-consuming, sometimes requiring months to detect one PPI. In the last several years, *large-scale* biological PPI experiments have been introduced to directly detect hundreds or thousands of protein interactions at a time. Yeast two-hybrid (Y2H) screens [32, 36, 75, 86] and protein complex purification detection techniques using mass spectrometry [23, 24, 32] are the two most widely used large-scale approaches. However, both methods suffer from high false positive and false negative rates [55]. For the Y2H method, this is due to insufficient depth of screening and misfolding of the fusion proteins. In addition, interaction between "bait" and "prey" proteins has to occur in the nucleus, where many proteins are not in their native compartment. The mass spectrometry based complex identification methods [23, 24, 32] may miss complexes that are not present under the given conditions. In addition, tagging may disturb complex formation and weakly associated components may dissociate and escape detections. In general, the resulting data sets are often incomplete and exhibit high false positive and false negative rates [15, 55, 99]. Consequently, even for well-studied model organisms, most true PPIs have not yet been discovered experimentally.

Computationally, protein-protein interaction networks can be conveniently modeled as undirected graphs, where the nodes are proteins and edges represent physical binding interactions. Initially, this graph is missing many edges (false negatives) and contains many incorrect edges (false positives). To complement and extend experimental methods, a variety of computational methods have been successfully applied to predict protein interactions. These approaches may be categorized on the basis of the types of data they considered when making predictions, as follows:

- Over-represented domain pairs or motif pairs observed in interacting protein pairs have been studied and used to infer PPIs. We provide more details of domain-domain interactions in Sect. 21.3. Structural information and sequence evidence about PPI interfaces has been used to predict potential PPIs [13, 21] as well.
- Various genomic methods infer protein interactions based on the conservation of gene neighborhood (Fig. 21.2), conservation of gene order, gene fusion events, or the co-evolution of interacting protein pair sequences [54, 82].
- An attractive alternative approach is to integrate various types of evidence from multiple sources in a statistical learning framework. A number of classification methods have been explored and multiple ways of using biological evidences have been studied in this framework [6, 8, 38, 60, 67, 71, 78, 96, 98, 101].
- High-throughput PPI experiments for elucidating protein-protein interactions have been applied to model organisms in recent years. Unfortunately the derived data sets are noisy and incomplete [55]. Multiple computational techniques have been proposed to improve the data reliability [5, 10, 84].

In the next sections, we describe in detail methods that fall into the latter three categories.

As mentioned above, interactions among proteins can take on many forms. Most previous computational works either predict direct physical interactions between proteins, or to identify if two proteins operate in the same complex, or to predict if two proteins are functionally linked to each other. The readers should keep this distinction in mind for the following methods. Qi et al. [66] performed a systematic comparison between these tasks and found that the task of identifying co-complex relationship seems to be easier than the other two tasks, with respect to the feature evidence they used.

21.2.1 Genomic Inference with Context

Accurate and large-scale prediction of protein-protein interactions directly from protein sequences is one of the important challenges in computational biology. Reviewed in [82] as "genomic inference methods" (including gene neighbor, gene fusion, and phylogenetic profile approach), this category uses genomic or protein context to infer functional associations between proteins.

Gene neighborhood: The idea of the gene neighborhood approach is shown in Fig. 21.2. We can see that genes $P1$, $P2$ and $P3$ are neighbors across three different genomes. From this association, we infer that their protein products are likely to associate with one another. The gene neighborhood approach provides strong signals for functional association between gene products within and across species [54], but this approach is arguably less well suited for specifically detecting physical interactions.

Gene fusion: The gene fusion approach [52], infers protein interactions from protein sequences in different genomes. It is based on the observation that some interacting proteins/domains have homologs in other genomes that are fused into one protein chain. Figure 21.3 gives an example of "gene fusion."

Phylogenetic profile: The phylogenetic profile method [65] is based on the observation that interacting proteins need to be present simultaneously in order to perform their functions. Therefore, the repeated co-occurrence of a pair of proteins across different organisms provides evidence that they interact. As shown in

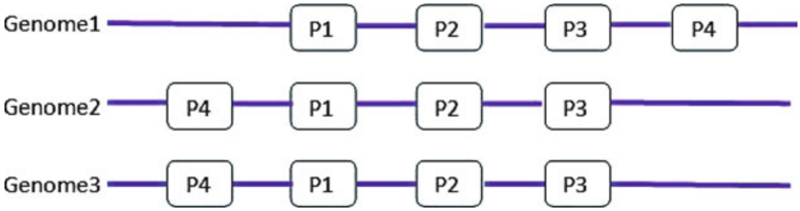


Fig. 21.2 PPI prediction by gene neighborhood approach (modified from Fig. 1 in [82])

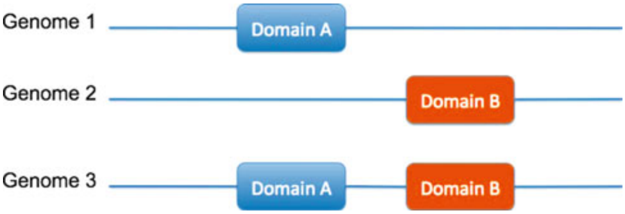
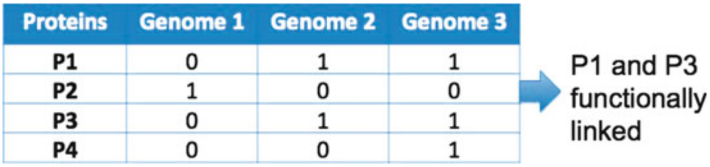


Fig. 21.3 PPI prediction by gene fusion (modified from Fig. 1 in [82])



Proteins	Genome 1	Genome 2	Genome 3
P1	0	1	1
P2	1	0	0
P3	0	1	1
P4	0	0	1

P1 and P3 functionally linked

Fig. 21.4 PPI prediction by phylogenetic profile strategy (modified from Fig. 1 in [82])

Fig. 21.5 PPI prediction by classification with multiple evidence

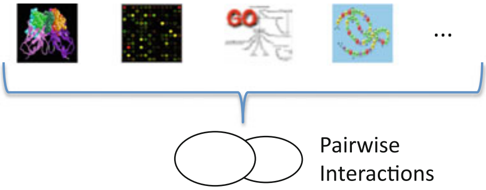


Fig. 21.4, a phylogenetic profile is constructed for each protein as an N-dimensional vector, where N is the number of genomes under consideration. The presence or absence of a given protein in a given genome is indicated with a 1 or 0 at each position in the profile. Proteins' phylogenetic profiles can then be linked using a bit-distance measure, with linkage indicating physically interaction or functional assocation [65, 82]. This approach can also be used for protein domains, where a profile is constructed for each domain.

21.2.2 Classification from Multiple Types of Evidence

Studies in this category make use of a classification algorithm to integrate diverse biological datasets (Fig. 21.5). A classifier is trained to distinguish between positive examples of truly interacting protein pairs and negative examples of non-interacting pairs. Many different research groups have independently suggested using supervised learning methods for predicting protein interactions. However, the data sources, approaches and the species they worked on have varied widely. According to these differences, we categorize previous works into four groups: supervised classifiers on protein pairs, kernel based network reconstruction, direct modeling of PPI data sets, and inter-species PPI prediction.

21.2.2.1 Supervised Classifiers on Each Protein Pair Separately

By transforming multiple biological data sources into a feature vector representing every pair of proteins, the task of predicting pairwise protein interactions can be formalized as a binary classification problem. Each protein pair is encoded as a feature vector where features may represent a particular information source such as related mRNA expressions, domain composition, or evidence coming from experimental methods. There are many possible ways to encode evidence sources into feature attributes and it is an important factor for the reliability of the computational predictions [66]. For instance, pearsons correlation values between two genes could be used as features on selected gene expression sets. Alternatively, feature attributes could describe how likely two proteins interact in other species [54].

A number of proposed methods belong to this group, including naive Bayes classifiers [38] , decision trees [101], kernel based methods [6, 96], random forests [51, 67], logistic regression [4, 51], and the strategy of summing likelihood ratio scores to predict PPI confidence in human [70, 71, 78] or in yeast [47]. Multiple classifiers were compared for PPI predictions in yeast [66]. Random forests and support vector machines (SVMs) were found to achieve the best performance among them.

These approaches used different types of data, different supervised classifiers and generally treated each protein pair independently for the interaction identification.

The popular STRING database [54] is a successful example of an application of this supervised learning methodology. The authors identified functionally associated protein pairs by computationally integrating known protein-protein associations, co-expression pairs, literature mining and pairs transferred across organisms. The resulting STRING database integrates and ranks predicted PPIs, by benchmarking them against a common reference set with the modified sum of likelihood approach. The most recent version of STRING [40] covers about 2.5 million proteins from 630 organisms. The authors claim that this provides the most comprehensive view of PPIs currently available.

Most of the above scoring methods use a set of likely true positives to train the predictive model. However, a single positive training set may be biased and not representative of true interaction space. To address this concern, Yu et al. [100] demonstrated a method to score protein interactions by using multiple independent sets of training positives to reduce the potential bias inherent in using a single training set. Defining negatives can also be problematic, since the absence of an edge in an observed network does not necessarily imply that the edge does not exist in the true network. Several studies attempt to define a set of high-confidence non-interacting proteins [39]; however, such methods are likely to yield their own biases [7]. Thus, the simpler approach of selecting negatives uniformly at random is generally preferred [6, 28, 68, 102].

21.2.2.2 Network Reconstruction with Kernel Methods

As mentioned above, multiple data evidence used for PPI predictions are in different formats (e.g., numeric values for gene expression, letter strings for protein sequences). A natural choice for this data integration task is kernel methods [6], which unify the data representation as special matrices called kernels (Fig. 21.6b). Kernel methods have been applied successfully on the protein interaction prediction tasks in recent years. The problem of PPI predictions could be framed as the following network reconstruction problem (Fig. 21.6). The input is a graph $G = (V, E, \bar{E})$, where V is a set of nodes representing each protein, and $E, \bar{E} \subset V \times V$ are sets of known edges and non-edges, respectively, corresponding to protein pairs that are known to interact or not. This PPI graph is represented as an adjacency matrix in Fig. 21.6a which contains known interactions (black boxes), known non-interactions (white boxes) and pairs with unknown status (gray boxes). In Fig. 21.6b, kernel methods build kernel matrices (graphs) based on features of proteins or protein pairs. The key question then is to reconstruct those "?" entries in the input PPI graph (gray boxes of Fig. 21.6a) based on the kernel graph(s) (Fig. 21.6b). Here we describe three interesting papers in this group.

Pairwise kernel between protein pairs: Ben-Hur et al. [6] and Gomez et al. [27] proposed the pairwise kernel approach to use a standard kernel method (such as SVM) for PPI predictions. Treating each protein pair as a data example, a pairwise kernel function computes the similarity between two pairs of proteins. Thus, with n proteins, the resulting kernel matrix (an example in Fig. 21.8b) contains n^4 entries. One way to construct such a kernel is to build them on top of an existing kernel between individual proteins. For example, given a kernel matrix K with each entry describing the inner product between two proteins, the pairwise kernel could be built for the four proteins in Fig. 21.8a as follows:

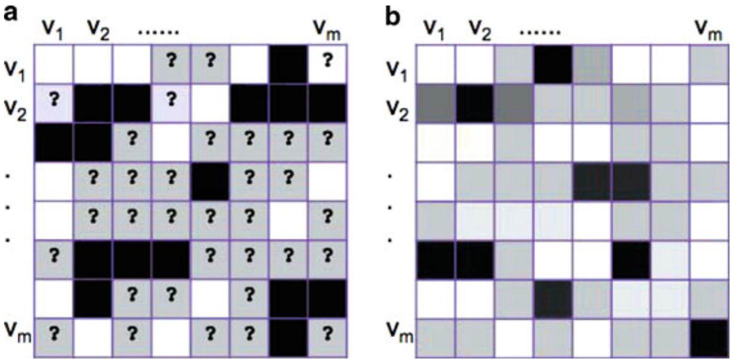


Fig. 21.6 PPI predictions through kernel methods (modified from Fig. 1 of [98]). (**a**) PPI network is represented as an adjacency matrix which includes: known interactions (*black boxes*), known non-interactions (*white boxes*) and pairs with unknown status (*gray boxes*). (**b**) Kernel matrix built from a certain feature evidence, with a *darker color* describing larger value

$$K'((v_1, v_2), (v_3, v_4)) = K(v_1, v_3)K(v_2, v_4) + K(v_1, v_4)K(v_2, v_3) \qquad (21.1)$$

The motivation is that protein pair (v_1, v_2) is similar to protein pair (v_3, v_4) if the two proteins v_1 and v_2 are similar to proteins v_3 and v_4, or vice versa. Later, Martin et al. [53] proposed a similar way to make use of protein properties for PPI prediction task, but with a tensor product kernel.

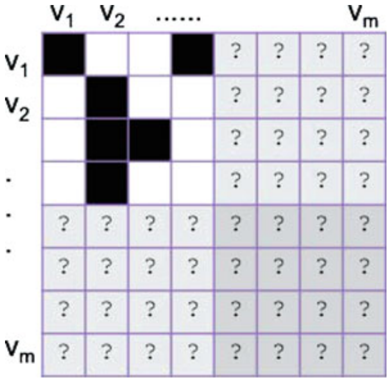
As a continuation of this work, the authors in [69] predicted co-complexed protein pair (CCPP) relationships using kernel methods from heterogeneous data sources. They show that a diffusion kernel [45, 83] based on random walks on the full network topology yields good performance in predicting CCPPs from protein interaction networks (for more details about this kernel, see Sect. 21.4.5) . In their setting of direct ranking, a diffusion kernel performs much better than the mutual clustering coefficient. Alternatively, when using SVM classifiers, a diffusion kernel performs much better than a linear kernel. One recent work from Vert et al. [91] explored a closely related approach called the "metric learning pairwise kernel" to convert the problem of direct inference based upon similarities between nodes joined by an edge on the PPI graph to the task of distance metric learning.

Note that the pairwise kernel strategy also belong to the group of methods in Sect. 21.2.2.1. Those methods use feature values to describe each protein pair. With an inner product between these features vectors, we could generate a pairwise kernel matrix. Of course, the way to calculate the kernel matrix in Eq. 21.1 is more general, since the pairwise kernel could incorporate data from individual proteins (using a pairwise kernel) and protein pairs.

Supervised reconstruction with a kernel between proteins: Because the computational cost for the above pairwise kernel is high, Yip et al. [98] and Yamanishi et al. [96] proposed to work directly with kernels defined on individual proteins. Given such a kernel K (between proteins) and a cutoff t, the method simply predicts interactions for each pair of proteins for which $K(v_i, v_j) \geq t$.

To make use of the training examples, supervised algorithms were presented to reconstruct the kernel matrix based on a sub-matrix of known interactions. Assuming that the sub-network of the adjacency matrix is totally known (as shown in Fig. 21.7), the goal is to modify the kernel similarity between proteins (as defined by the kernel) to some values that are more consisitent with the partial sub-matrix. Subsequently, simple thresholding is performed on the resulting similarity values to predict PPIs [98]. Yamanishi et al. [96] presented a method in this style to infer protein interaction networks using a variant of kernel canonical correlation analysis (originated from spectral clustering theory). The goal was to identify features from the input kernel (built from the genomic/proteomic evidence) and features from the diffusion kernel that were derived from the known PPI submatrix, so that two features have the highest correlation under certain smoothness requirements.

Kernel matrix completion: Similar to the above supervised network reconstruction, Kato et al. [42] also assume a partially complete adjacent matrix (Fig. 21.7). They formulated supervised network inference as a kernel matrix completion problem, where the inference of edges boils down to estimation of missing entries of a kernel matrix. The goal is to make the resulting matrix closest to a spectral variant



	V_1	V_2	......					V_m
V_1	■			■	?	?	?	?
V_2		■			?	?	?	?
		■	■		?	?	?	?
.		■			?	?	?	?
.	?	?	?	?	?	?	?	?
.	?	?	?	?	?	?	?	?
	?	?	?	?	?	?	?	?
V_m	?	?	?	?	?	?	?	?

Fig. 21.7 PPI predictions by the supervised network inference (modified from Fig. 1c of [98]). Partial complete adjacency matrix required by the supervised reconstruction approach, which needs complete knowledge of a submatrix (*upper-left*)

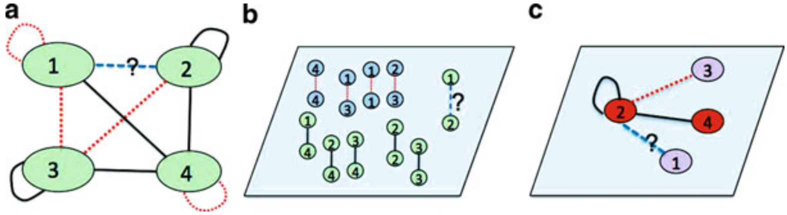


Fig. 21.8 Global and local modeling for PPI network reconstruction (modified from Fig. 2 of [98]). (**a**) An interaction network, with *solid black lines* representing known interactions, *red dotted* edges representing known non-interacting edges and *blue dashed lines* representing those protein pairs with unknown interaction status. (**b**) Global model based on pairwise kernel approach, where each edge is treated independently. (**c**) Local model for protein v_2. Different node colors indicate distinctive evidence status, for instance, different cell compartments that the proteins reside in

of the kernel matrix as measured by the KL (Kullback-Leibler) divergence. An expectation-maximization algorithm is proposed to simultaneously infer the missing entries of the adjacency matrix and the weights of multiple datasets (a weight is assigned to each type of dataset and thereby to select informative ones). The algorithm iteratively searches for the filled adjacent matrix that is closest to the current spectral variant of the kernel matrix, and at the same time, the spectral variants of the kernel matrix which is closest to the current filled matrix. When convergence is reached, the predictions are thresholded from the final complete adjacency matrix.

Local model: Each of the above approaches builds a global model to predict new edges over the network based on the partial knowledge of the network to be inferred (Fig. 21.8b). This single model may not be able to separate all cases of interacting pairs from non-interacting ones, if there are different subgroups of interactions [98]. For instance, protein pairs involved in transient interactions may use a very different strategy compared with those involved in protein complexes. These two types of interactions may belong two separate subgroups that cannot be fitted by one single model.

Accordingly, Bleakley et al. [8] introduce a novel method that uses a local model to allow for flexible modeling of subgroups of interactions. A local model is built for each protein, using the known interactions and non-interactions of this protein as the positive and negative examples. The resulting classification rule predicts edges associated with a single protein. Thus, each pair of proteins receives two predictions, each from the local model of either protein. In Fig. 21.8c, the method built a local model for protein v_2. Because node v_1 is similar to node v_3, this local model classified pair (v_2, v_1) as negative. Since each node has its own local model, the approach only needs a kernel defined on proteins, rather than a kernel between pairs of proteins.

Local model with training set expansion: The accuracy of computational techniques proposed for PPI network reconstruction is consistently limited by the small number of high-confidence examples. Specifically, for the local model approach, the uneven distribution of positive examples across the potential interaction space, with some objects having many known interactions and others few, makes it hard to predict new interaction partners for those proteins having very few known interactions reliably. To address this issue, Yip et al. [98] proposed two semi-supervised learning methods by augmenting the limited number of gold-standard training instances with carefully chosen and highly confident auxiliary examples.

- The first method, *prediction propagation* is similar to self-training methods [79] described in the the machine learning community. This method uses highly confident predictions from one local model as the auxiliary examples of another. This propagation strategy uses the learning from information-rich regions in the training network to help make predictions in information-poor regions.
- The second method, *kernel initialization*, takes the most similar and most dissimilar proteins of each protein in a global kernel (between proteins) as the auxiliary examples. Similar to prediction propagation, adding these new examples into the training sets boosts the performance of the local modeling approach.

21.2.2.3 Inter-species PPI Prediction

All of the above studies aim to predict PPIs within a single organism (termed *intra-species PPI prediction*), with most studies focusing on yeast or human. Recently, researchers have begun to extend computational methods to predict PPIs between species (termed *inter-species PPI prediction*).

Of particular interests are host-pathgen PPIs. For any host-pathogen system, it is important to understand the mechanism by which a pathogen can infect its host. One method of infection is via protein interactions, where pathogen proteins target host proteins (as described in Fig. 21.9). Developing computational methods that identify which PPIs enable a pathogen to infect a host has significant implications in identifying potential therapeutical targets.

Davis et al. [16] studied ten host-pathogen protein-protein interactions using structural information with a comparative model: the host/pathogen protein pairs that share similarity to protein complexes with known structures are used to build 3-D structural models of putative complexes, and the modelled pairs are then filtered by functional and genomic experimental information. The technique was applied

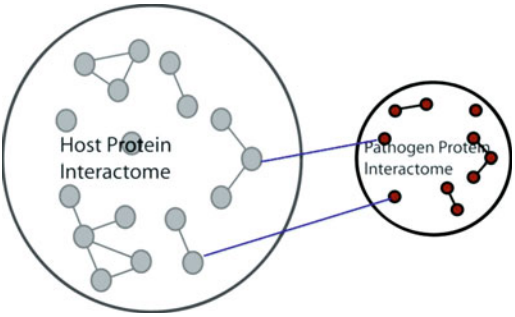


Fig. 21.9 Protein-protein interactions in host-pathgen systems (modified from Fig. 1 of [87])

to ten pathogens and assessed by three independent computational procedures. The results suggest that this method is complementary to experimental efforts in elucidating networks of hostpathogen protein interactions.

Later, Tastan et al. [87] extended the supervised learning framework to predict PPIs between HIV-1 viruses and human proteins. A random forest based classifier was used to integrate multiple biological data types, achieving state-of-the-art performance for this task.

Similar to host-pathgen PPI, several recent papers identify interactions between drugs and target proteins. This is a key area in genomic drug discovery. The authors in [95] formalized the drug-target interaction inference as a supervised learning problem on a bipartite graph, where the model extended the metric embedding approach [1] to integrate chemical and genomic spaces into a unified space.

21.2.3 Modeling Experimental PPI Data Sets Directly

Genome-wide, high-throughput PPI experiments for elucidating protein-protein interactions have proven to be one of the most important tools in recent years. However the quality of currently available PPI data sets is unsatisfactory, which limits its usefulness to some degree. A crucial step in analyzing proteomics PPI data is to separate the subset of credible interactions from the background noise. Various computational techniques have been proposed for inference of reliable protein-protein interactions directly from experimental interaction results. In the following, several interesting ones are covered.

Von Mering et al. [55] were among the first to discuss the problem of accurately inferring protein interactions from high-throughput data sources. The proposed solution [55], which used the intersection of direct high-throughput experimental results, achieved a very low false positive rate. However, the coverage was also very low. Less than 3% of known interacting pairs were recovered using this method.

Later, Bader et al. [5] applied logistic regression to estimate the posterior probability that a pair of proteins will interact. Only statistical and topological descriptors were used to predict the biological relevance of protein-protein interactions obtained from high-throughput PPI screens for yeast. Other evidence, such as mRNA expression, genetic interactions and database annotations, were subsequently used to validate the model predictions. They demonstrated that it is possible to define a quantitative confidence measure based entirely on screening statistics and network topology. The main assumption underlying the confidence measure is that nonspecific interactions are highly likely to be technology-specific [5]. This type of analysis is essential for analyzing the growing amount of genomic and proteomics interation data in model organisms.

Aiming to improve the quality of experimentally available PPI data by identifying erroneous datapoints from PPI experiments, Sontag et al. [84] described a probabilistic approach to estimate errors in yeast-two-hybrid experiments, considering both random and systematic errors. The systematic errors arise from limitations of the Y2H experimental protocol: ideally the reporting mechanism in Y2H should be activated if and only if the two proteins being tested truly interact, but in practice, even in the absence of a true interaction, the reporter may be activated by some proteins – either by themselves or through promiscuous interaction with other proteins. The authors described a probabilistic relational model that explicitly models these two types of errors. They use Markov chain Monte Carlo algorithms for inference. In constrast to previous work, which often models Y2H errors as being independent and random, experimental results showed that this approach could make better use of the available experimental data.

Currently no method exists to systematically and experimentally assess the quality of individual interactions reported in interaction mapping experiments. Braun et al. [10] developed an interaction tool kit consisting of four complementary, high-throughput protein interaction assays and provided a standardized confidence-scoring method. Based on positive and random reference sets consisting of well documented pairs of interacting human proteins and randomly chosen protein pairs, a logistic regression model was trained to combine the assay outputs and calculate the probability that any newly identified interaction pair is a true biophysical interaction once it has been tested in the the four high-throughput PPI assays. This approach allows a systematic and empirical assignment of confidence scores to all individual protein-protein interactions from high throughput interation experiments.

The above approaches have considered protein pairs independently when inferring the presence of PPIs. In contrast, Jaimovich et al. [37] considered the neighborhood interaction pairs together and employed a *relational Markov random field* approach for collective inference of PPIs in yeast. The basic idea is shown in Fig. 21.10:

In this paper [37], the authors view the PPI prediction task as a relational learning problem, where observations about different entities are not independent. The method exploits relational probabilistic models to combine multiple types of features, including protein attributes (e.g., localization of proteins) and protein-protein interactions (e.g., experimental interaction assays). The results demonstrated that modeling the dependencies between interactions leads to significantly better predictions. However, due to the model complexity and the difficulties during inference, this model can currently be applied only to a small set of proteins.

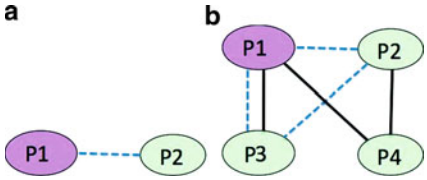
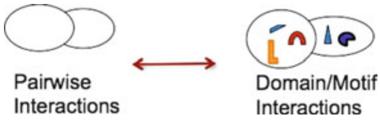


Fig. 21.10 Improve PPI prediction with dependencies between interactions (modified from Fig. 1 in [37]). (**a**) A possible interaction between proteins P1 and protein P2. They are localized in different cellular positions (indicated with *purple and green colors*). (**b**) Two additional proteins P3 and P4 provide extra dependency evidence. *Dashed line* represents functional association from indirect evidence and *solid line* describes interactions from experimental interaction sets. The combined evidence gives more support to predict that P1 and P2 interacts

Fig. 21.11 Prediction of domain-domain interactions



21.3 Prediction of Domain-Domain Interactions

Many of the experimental and computational approaches described above address the question, "Do these two proteins interact?" In practice, *how* the proteins interact is also of great interest. Protein interactions occur through physical binding of small regions on the surface of proteins. Therefore, insights into the mechanism whereby a protein carries out its function can be obtained by identifying the interaction site where protein binding takes place. Moreover, detailed knowledge about the binding sites at which an interaction takes place can provide insight into the causes of human disease as well as a starting point for drug design [92]. Unfortunately, this type of information is not typically provided in a protein interaction graph and is not revealed by high-throughput experimental methods.

A protein may contain a single domain or multiple domains, each one typically associated with a specific function [88]. The combination of domains determines the function of the protein, such as its subcellular localization and the interactions it is involved in [34]. There exists a certain degree of conservation in the interaction patterns between similar proteins and domains. It has been found that close homologs almost always interact in the same way [81]. Thus, it is interesting to find out what domains are responsible for binding.

Currently little useful data is available from major databases with respect to relations on the domain level [63]. This lack of data makes computational prediction of domain-domain interactions very important. A series of computational approaches have been developed to predict which domains in a protein pair interact given a set of experimental protein interactions [82]. Domain interactions extend the functional significance of proteins and provide a more detailed view of the protein-protein interaction network (Fig. 21.11).

Inferring interactions between domains from protein-protein interactions is a challenging task. Various methods have been proposed to predict domain interactions from protein-protein interaction graphs. Most methods begin by annotating protein sequences with domains that can be defined by Pfam, CDD, or other domain databases. The models are typically trained with certain known protein interactions to identify domain-domain interaction pairs. The predicted domain interactions can be evaluated using structural data or by high quality interaction sets. Moreover, the resulting domain interactions can in turn help in predicting protein-protein interactions. It is worthwhile to mention that some of the approaches mentioned in the last section for protein interaction prediction, such as the sequence co-evolution or phylogenetic profiles (reviewed in [63]) are also applicable to domain interaction prediction [82]. In addition, the following section introduces several methods specifically designed to predict domain-domain interactions from protein interaction data.

Inferences on the interactions among domains can be made by analyzing the domain composition of a set of proteins and their interaction networks.

Association method: A characteristic domain or structural motifs can be used to distinguish interacting proteins from non-interacting. Association methods [30, 82, 85] use different classifiers for this purpose, and some of them are tuned specifically to identify domains responsible for protein interactions. *Correlated domains* are pairs of domains that are found together more often than expected by chance in known PPI pairs. An association method may predict that two proteins interact if they contain correlated domains, one from each protein, whose association value is greater than a predefined threshold. Because some domain pairs can be found quite often in protein interacting pairs, this simple assocation method can be quite successful in identifying novel PPIs.

An examplar case is given in Fig. 21.12a. Domain pair (x, a) is the most abundant in all four interacting protein pairs (blue lines) compared with other domain-domain

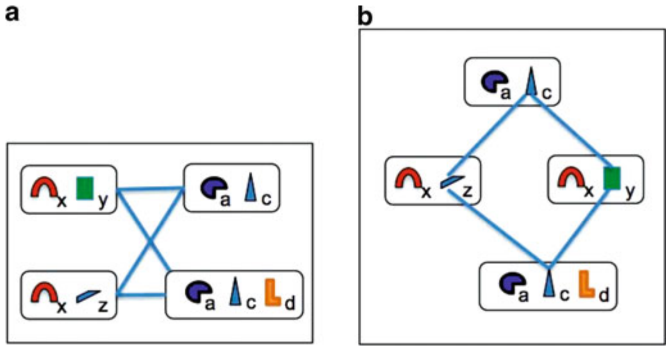


Fig. 21.12 Two methods to predict domain-domain interactions from PPIs. (**a**) Association method. The domains x and a are predicted to interact due to the abundance of domains x and a in protein interaction pairs, shown as the *blue line*. (**b**) As the same PPI dataset in (**a**), that the actual domain interactions (*blue lines*) do not include domains x and a. This shows that accounting for other domains in a protein pair, in addition to x and a, can result in alternative domain interaction predictions

pairs. Taking the domain combination pair as a basic unit, these methods use their frequencies in the interacting and non-interacting sets of protein pairs, for deriving novel protein interactions. For example, Sprinzak et al. [85] use the following score, computed from protein interaction data, to find correlated domains:

$$S(d_m, d_n) = \frac{I_{mn}}{N_{mn}} \tag{21.2}$$

where I_{mn} is the number of interacting pairs that contain (d_m, d_n), and N_{mn} is the total number of protein pairs that contain (d_m, d_n).

Dyer et al. [20] extended this idea for identify domain interactions in host-pathogen systems. They integrate a number of public intra-species PPI datasets with protein-domain profiles for predicting and studying host-pathogen PPI networks. The model used intra-species PPIs and protein-domain profiles to compute statistics on how often proteins containing specific pairs of domains interact. These statistics can then be used to predict inter-species PPIs in host-pathogen systems.

Maximum Likelihood Estimation: One drawback of the association method is that it ignores other domain-domain interaction information between the protein pairs and, thus, does not make full use of all of the available information. As in Fig. 21.12a, if domains x and a do not appear in any other proteins, then in the association method this pair is assigned the association score $S(x, a) = 4/4 = 1$. This method ignores other domain-domain interactions among domains b, c, y and z. To infer a domain-domain interaction, other related domain-domain interactions should be taken into account (as shown in Fig. 21.12b). To do so, interactions among other proteins containing domains b, c, y or z must be included, and thus, more domains and proteins are involved. Iterating this process, eventually all proteins and all domains are related and need to be taken into account. In addition, the association method ignores experimental errors (normally quite high in current experimental PPI sets) and treats the observed interactions as real interactions. This noise may lead to the impossibility of having a pattern of domain interactions that is compatible with the protein-protein interaction map.

To address the above two issues, Deng et al. [18] develop a global approach using a maximum likelihood estimation (MLE) method that incorporate all available proteins and domains, as well as experimental errors. They used yeast two-hybrid protein interaction data and treated protein sequences as "bags of domains." The model estimates the probabilities of interactions between every pair of domains. Treating protein-protein interactions and domain-domain interactions as random variables, the two basic assumptions are (1) that two proteins interact if at least one pair of domains of the two proteins interacts and (2) interactions between different domain pairs are independent. Thus, the probability of a potential interaction between a protein pair (i, j) is

$$P(P_{ij} = 1) = 1 - \prod_{(d_m, d_n) \subset (P_i, P_j)} (1 - \lambda_{mn}) \tag{21.3}$$

where λ_{mn} denotes the probability that domain d_m interacts d_n. The expectation maximization (EM) algorithm is used to find maximum likelihood estimates of unknown parameters by finding the expectation of the complete data consisting of observed and unobserved data in two iterative steps. Here the observed data includes protein-protein interactions and the domain composition of the proteins, and the unobserved data includes all putative domain-domain interactions [82].

The above methods may preferentially identify promiscuous domain interactions, because they focus on those that occur with the highest frequency. Methods are need to detect the low-propensity, high-specificity domain interactions. Thus, Riley et al. [73] proposed the domain pair exclusion analysis (DPEA) method to extend the MLE approach. Riley et al. are specifically interested in extending beyond single proteome prediction to infer domain interactions from the incompletely mapped interactomes of multiple organisms. Their appoach employs a likelihood ratio test to assess the contribution of each potential domain interaction to the likelihood of a set of observed protein interactions from the incomplete interactomes of multiple organisms.

Similarly, Iqbal et al. [35] address the problem of predicting protein domain interactions by using belief propagation, which is a powerful message passing algorithm for probablistic inference. The input to their algorithm is an interaction map among a set of proteins, and a set of domain assignments to the relevant proteins. The output is a list of probabilities of interaction between each pair of domains. The method is able to effectively cope with errors in the protein-protein interaction dataset and systematically resolve contradictions.

Hypothesis test: Nye et al. [61] proposed a statistical method to test the null hypothesis that the presence of a particular domain pair in a protein pair has no effect on whether two proteins interact. The procedure calculates a statistic for each domain pair which takes into account experimental errors and the incompleteness of the dataset. The background distribution is simulated by shuffling domains in proteins so that the network of protein interactions remains fixed. The domain pair with the lowest p value is deemed most likely to interact. The authors point out that, for the majority of test cases, random domain prediction outperforms all methods tested, indicating the low accuracy of all prediction methods of domain interactions.

A set cover approach: Later, Huang et al. [33] proposed an interesting model to map the relationship between interactions of proteins and their corresponding domain architectures to a generalized set cover problem. Figure 21.13 gives a schematic explanation of the set cover approach. Set Y represents all potential protein pairs, and set X describes all known protein interaction pairs. $F = \{S_i, 1 \leq i \leq t\}$ is a family of subsets of Y. The general set cover problem is to find a subset C of F to cover X, such that $X \subseteq \cup_{S \in C} S$. Often, C is required to satisfy certain conditions. In this case, F is the set of all domain pairs (d_m, d_n). Specifically if a protein interaction pair (P_i, P_j) contains domain pair (d_m, d_n), then (P_i, P_j) belongs to the subset of (d_m, d_n). The goal is to find the collection C to cover X, where C is a subset of F and contains all the domain pairs present in the interaction network. The authors applied a greedy algorithm to identify sets of domain interactions which explain the presence of protein interactions to the largest degree of

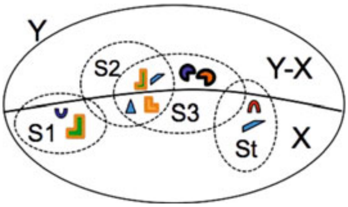


Fig. 21.13 A set cover approach to predict domain interactions from PPIs. Y set represents all potential protein pairs. X set includes all known protein interaction pairs

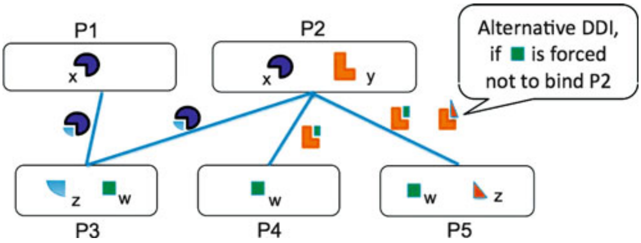


Fig. 21.14 Basic idea to predict protein interaction sites with the InSite method [92]. This figure is modified from Fig. 1 in [92]

specificity. Using domain and protein interaction data from *S. cerevisiae*, they claim that this model enables prediction of previously unknown protein interactions.

Prediction with additional information: Recently, researchers started to combine PPIs with a variety of additional types of evidence to predict domain interactions. For example, Wang et al. [92] propose a learning method, called InSite, to predict specific regions (domains or motifs) where protein-protein interactions take place. The input includes a library of conserved sequence motifs or domains, a set of protein-protein interactions, and any available indirect evidence on protein-protein interactions and motif-motif interactions, such as expression correlation, gene functional annotation, and domain fusion. InSite makes predictions at the level of individual protein pairs, in a way that takes into consideration the various alternatives for explaining the binding between this particular protein pair. Specifically, this method integrates multiple biological data sets and generates predictions in the form of 'Motif Y on protein P2 binds to protein P5' (as shown in Fig. 21.14). In contrast to previous methods, which predict bindings between pairs of motif types, InSite makes predictions of interactions of particular occurrences of two motifs. Thus, InSite may give the same motif pair different interaction confidences, depending upon the sequence context and the local neighborhood of the PPI network (Fig. 21.14). This approach provides a principal way to integrate all available biological evidence. It also treat PPIs from multiple assays differently, since some of them are noisy and some are indirect.

As above, we briefly discuss several important approaches to the task of identifying interacting and/or functionally linked domain pairs. These methods exhibit varying levels of success; however, they usually assume that domains interact independently, which is a limitation. Also part of the prediction errors come from incomplete domain assignments, insufficient coverage of domain databases and limited searching ability of domain profiles. In addition, domain interactions are

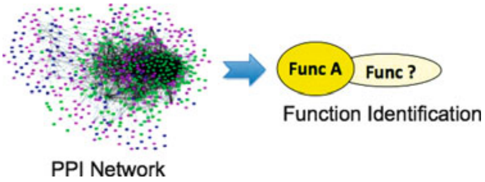


Fig. 21.15 Prediction of protein function from PPI networks

predicted from protein interactions, whose available data is incomplete and noisy at the current stage [82].

There exist a number of important problems related to the domain-domain prediction from PPIs, including the interaction sites' prediction or the docking task. Since they are beyond the scope of this chapter, interested audience could refer to the review paper Zhou et al. [103] for the first task and Ritchie et al. [74] for understanding the second: docking problem.

21.4 Prediction of Protein Function from PPI Networks

Proteins are involved in practically every function performed by a cell. However, despite the availability of large amounts of DNA and protein sequence data, the biological function is still unknown for a large proportion of sequenced proteins. Moreover, a given protein may have more than one function, so many proteins that are known to be in one functional class may have as yet undiscovered functionalities [97].

Inferences about function can be made via protein-protein interactions because protein interactions directly contribute to protein function. The premise is that the unknown function of a protein may be discovered through its interaction partners. Besides protein interaction evidence, the function of an unannotated protein can be predicted through various other data sets, including sequence homology, phylogenetic profiles, gene expression and so on. Combining multiple data sources together for protein function prediction is an interesting computational problem [11, 64, 89].

Here we focus on reviewing computational approaches that use protein-protein interaction evidence for protein function inference. It is worth mentioning that the interaction partners for a protein may belong to different functional categories. The problem of functional assignments in the complex protein network of within-function and cross-function interactions remains a difficult task [80].

Previous efforts in this area can be grouped into six categories, which are described in the following sections.

21.4.1 Simple Statistical Test

The basic assumption of functional annotation is that proteins which lie closer to one another in the PPI network are more likely to have similar functions. Thus, a simple statistical test can be used to assign functions to proteins based on the functions of their interaction partners.

For instance, Schwikowski et al. [77] proposed the neighborhood-counting method to assign k functions to a protein by identifying the k most frequent functional labels among its interacting partners. This strategy is simple and effective, but the full topology of the network is not taken into account in the annotation process, and no confidence scores are created for the annotations.

Another typical technique, referred to as the chi-square method [31], assigns k functions to a protein with the k largest chi-square scores. For a protein p, each function f is assigned a score $\frac{(n_f - e_f)^2}{e_f}$, where n_f is the number of proteins in the n-neighborhood of p that have the function f. The value e_f is the expectation of this number based on the frequency of f among all proteins in the network [80].

Recently Lee et al. [48] extended the neighborhood-counting [77] method to make network-based prediction of loss-of-function phenotypes in Caenorhabditis elegans. For a given phenotype, each gene in the worm proteome was ranked-ordered by the sum of its linkage weight (log-likelihood score of the gene interaction edge) to the "seed" set of genes already known to show that phenotype. The high-scoring genes are most likey to share the given phenotype.

In general, these simple methods lack a systematic mathematical model.

21.4.2 Graph Topoplogy

Researchers have also explored a variety of graph algorithms for protein functional inference [41,58,90]. For instance, Vazquez et al [90] and Karaoz et al. [41] exploit the global topological structure of the interaction network for functional annotation. The basic idea is described with a simple schematic example in Fig. 21.16. This is a subgraph of the protein interaction network in the yeast *Saccharomyces cerevisiae*, with yellow nodes representing unannotated proteins and blue nodes representing annotated ones (the associated functions are listed as numbers in brackets adjacent to the nodes). Given one of these proteins with unknown functions, a simplified version of the method (proposed in [77]) would predict the function that appears most often in the neighbor proteins of known function. This approach would lead to the following classification result (from top to bottom): P3 (2), P4 (3,4,10) and P5 (12). By contrast, graph algorithms such as the one proposed by Vazquez et al [90], would also consider the interactions among unclassified proteins. Taking into account the interactions among the three unclassified proteins, one more iteration of the "majority rule" would lead to the following classification: P3 (2,4), P4 (3,4,10) and P5 (12). Thus, this extended method determined another possible function for P3.

The approach proposed in [90] assign proteins to functional classes so as to maximize the number of edges that connect proteins (unannotated or previously annotated) assigned with the same function. Precisely, they maximize

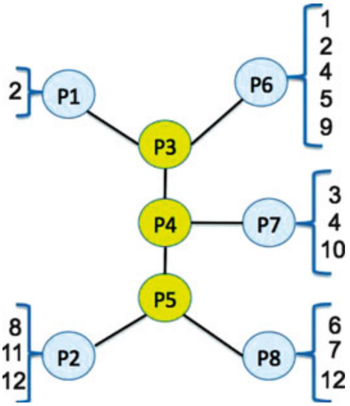


Fig. 21.16 Functional annotation from graph algorithm on PPI networks. Modified from Fig. 1 in [90]. This shows a subgraph of the protein interaction network of the yeast Saccharomyces Cerevisiae. Proteins in *yellow* are unannotated (unknown function); the others are classified proteins (functions in *brackets*)

$$\sum_{(i,j)\in E'} \delta(\sigma_i, \sigma_j) + \sum_{i\in V} h_i(\sigma_i) \tag{21.4}$$

where E' is the set of edges between two unannotated proteins, δ is a function that equals 1 if $x = y$ and 0 otherwise, V is the set of nodes (proteins), and $h_i(f)$ denotes the number of neighbors of protein i previously annotated with function f. The first term in the optimization criterion accounts for unannotated proteins, whereas the second term concerns the interactions between unannotated and previously annotated proteins. This optimization problem can be generalized to the computationally hard problem of minimum multiway cut. The authors solved it heuristically using simulated annealing in [90].

Karaoz et al. [41] additionally consider the case where edges in physical interaction networks are weighted using gene expression data. The approach is a generalization of the well-studied multiway k-cut problem. The authors apply a local search strategy in which the state of the vertex is changed according to the majority of the states of its neighbors. Similarly, Nabieva et al. in [58] developed a network flow algorithm that exploits the underlying structure of protein interaction maps in order to predict protein function. Unlike [41, 90], this method takes advantages of both network topology and a particular measure of locality.

21.4.3 Graph Clustering

Clustering on protein interaction networks can also be used to predict protein function. For example, Samanta and Liang [76] proposed a network-based statistical measure to represent how many common partners two proteins share. They then use this statistic to hierarchically cluster the proteins in the PPI network. The key idea is that two proteins that share a large number of common partners likely have close functional associations. Arnau et al. [3] also applied hierarchical clustering in the protein-protein interaction network to find functionally consistent clusters. Their similarity measurement is derived from the shortest distance between two proteins in the network. Unlike typical graph clustering, Airoldi et al. explored a generative style of clustering [2]. The authors used a latent mixture membership approach to model the protein-protein interaction network. This approach transforms the function prediction objective into learning of the latent groups.

Sharan et al. [80] recently reviewed current computational approaches on functional annotation of proteins in the context of the protein interaction networks. They split the related papers into two types: (1) direct annotation schemes, which infer the function of a protein based on its connections in the network, and (2) module-assisted schemes, which first identify modules of related proteins and then annotate each module based on the known functions of its members. Methods we cover in other subsections belong to the "direct scheme" category. The current subsection only briefly introduces module-based (we call "graph clustering" based) methods which utilized the modularity assumption of PPI networks. There exist a number of ongoing work that explore this category of strategies for protein function annotation. Readers interested should refer to the overview paper [80] for details. Basically, such methods first attempt to identify coherent groups of genes and then assign functions to all genes in each group. The module-assisted methods differ mainly in their module detection techniques, which include graph clustering, hierarchical clustering, clustering based on network topology, etc. Once a module is obtained, simple methods are usually used for function prediction within the modules.

21.4.4 Probabilistic Propagation on Belief Networks

Although there exist multiple functional classes, we can approach the functional annotation task one fuction at a time. Figure 21.17 gives a schematic illustration of this case. For a certain functional class, the proteins assigned this function are labeled "1". The proteins which are known to not have this function are labeled "0". The remaining nodes are marked "?". With this assignment, the protein-protein interaction graph in Fig. 21.17 can be treated as a probabilistic belief network of function annotations. A number of probabilistic approaches to protein function

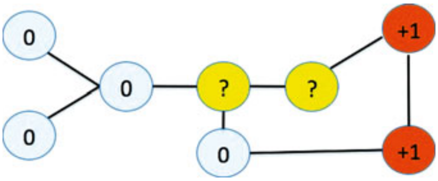


Fig. 21.17 A schematic illustration of the function prediction task on a protein network. Modified from Fig. 1 in [89]. The task is to predict labels of unannotated proteins marked as "?". For a specific functions proteins having that function are labeled with "1" or other wise "0"

prediction have been suggested. Most such approaches have relied on a Markovian assumption, namely, that the function of a protein is independent of all other proteins given the functions of its immediate neighbors [80] . This global approach takes all the network interactions and the functions of known proteins into consideration, propagating function labels from annotated proteins to unannotated proteins [17, 19, 49, 50].

The Markovian assumption naturally leads to a Markov random field (MRF) model, which was proposed by Deng et al. [19]. In this paper, an MRF was used to assign functions to unknown yeast proteins, with a probability representing the confidence in the prediction. Each protein node is assigned a random variable, with states corresponding to functional annotations in this setting. Thus, the interaction between two known proteins can be classified into one of the three groups: (1,1), (1,0) and (0,0), where numbers describe the involved proteins' functional annotation. The joint belief can then be represented with a Gibbs distribution by considering the classification of all proteins,

$$Pr(X|\mathit{PPInet}) = \frac{\mathit{exp}[-U(x;\theta)]}{Z(\theta)} \tag{21.5}$$

where

$$U(x;\theta) = -(\alpha N_1 + \beta N_{11} + \gamma N_{10} + \kappa N_{00}) \tag{21.6}$$

$U(x;\theta)$ represents the potential function of the PPI network given a functional configuration of all proteins $X = (x_1, \ldots, x_N)$ (discrete states). N_1 is the number of proteins for class "1," and $N_{ll'}$ is the number of protein interactions between category l and l' in the network. $\theta = (\alpha, \beta, \gamma, \kappa)$ are parameters, where κ is set equal to 1. $Z(\theta)$ is the normalization constant (called the *partition function*), which is calculated by summing over all the configurations,

$$Z(\theta) = \sum_x \mathit{exp}[-U(x;\theta)] \tag{21.7}$$

Inference in this model is computationally hard. Deng et al. [19] use a quasi-likelihood method to estimate the parameters θ. The posterior probability that an unknown protein has the function of interest given the annotations of its neighbors $P(x_v = 1|x_{N(v)})$ was calculated with a Gibbs sampler.

Letovsky and Kasif [50] assumed a binomial model for local neighbors of a protein annotated with a given term. Also using the MRF propagation this algorithm assigns probabilities for proteins' functional annotation in the network using loopy belief propagation. Leone et al in [49] proposed a belief propagation method on PPI networks in a similar framework.

Later, Wu et al [93] proposed a related probabilisitic model to annotate functions of unknown proteins on PPI networks. Their model is an implicit MRF model that considers all the functions in a single model. This approach allows the model to capture correlations among protein functions. The authors used the conditional distribution and presented a maximum likelihood formulation of the problem. The time complexity of the corresponding learning and inference algorithms is linear in the size of the PPI network.

Mostafavi et al [57] adopted a variation of the Gaussian field label propagation algorithm for gene function prediction. Like the methods described above, this method assigns a score to each node in the network. This score reflects the estimated degree of association that the node has to the seed list defining the given function. The scores can be thresholded to make predictions. Unlike previous approaches using MRFs, the Gaussian field algorithm has a well-defined solution and can be efficiently computed.

21.4.5 Kernel Method

Kernel machines have been applied extensively for discovering functionally similar proteins within interaction networks. This approach has the ability to integrate multiple types of evidence for functional predictions. For instance, Lanckriet et al. [46] and later Tsuda et al. [89] represent each data type using a matrix of kernel similarity values. These matrices are then combined by learning optimal relative weights for the different kernels.

Here we briefly describe how protein-protein interaction data can be used by a kernel method [89]. Normally, a diffusion kernel [45, 83] is calculated on the graph of proteins connected by interactions. The diffusion kernel is a general method for computing pairwise distances among all nodes in a graph, based on the sum of weighted paths between each pair of nodes. Assume that A is the $n * n$ adjacency marix of a graph, and D is the $n * n$ diagonal matrix such that D_{ii} is the node degree of i-th node. The graph Laplacian matrix is defined as $L = D - A$. The diffusion kernel [45, 83] is then defined as

$$K = exp(-\beta L) \tag{21.8}$$

where the diffusion parameter $\beta > 0$ determines the degree of diffusion. This kernel can be interpreted in terms of a "lazy" random walk for sufficiently small β. At each step, the next node is randomly chosen from the neighbor nodes according to the transition probabilities. One can also stay at the same node (which is why the random walk is called "lazy". The kernel value K_{ij} is equivalent to the probability that a random walk starting from i will stay at j after infinite time steps. Figure 21.18 shows the actual values of diffusion kernels with one possible β. When β is large enough, the kernel values among distant nodes can capture the long-range relationships between proteins [89]. Diffusion kernels offer several benefits: (1) these kernels consider similarities among all protein pairs on the graph, not just immediate neighbors, (2) node degrees are taken into account in the kernel calculations, and (3) the parameter β is relatively easy to tune and has a clear meaning.

Lanckriet et al. [46] (and many others) used a diffusion kernel [45, 83] to summarize PPI graph evidence for functional predictions. Later, Tsuda and Noble

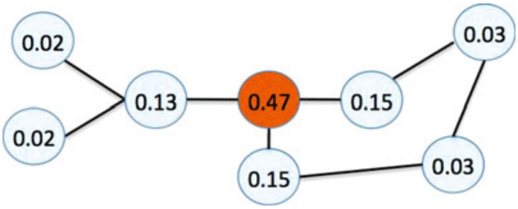


Fig. 21.18 Actual values of the diffusion kernel for one parameter setting of diffusion parameter β. Modified from Fig. 2 in [89]. Each value on a node shows the kernel value between the node and the central node (*orange* node). The kernel values diffuse through the nodes on the graph

[89] proposed a locally constrained variant of the diffusion kernel. They showed that computing the diffusion kernel is equivalent to maximizing the von Neumann entropy, subject to a global constraint on the sum of the Euclidean distances between nodes. This global constraint allows for high variance in the pairwise kernel distances. Thus, the authors proposed an alternative, locally constrained diffusion kernel and demonstrated that the resulting kernels allow for more accurate support vector machine predictions of protein functional classifications from the metabolic and protein-protein interaction networks.

21.4.6 Functional Identification Toward Annotation Taxonomy

The above two subsections handle the task of protein function prediction as multiple binary classications, where the methods treat each function at a time and make predictions for each term independently.

A more general approach to protein function prediction uses labels that follow a directed acyclic graph taxonomy as defined by the Gene Ontology (GO) [14]. The GO defines a set of terms to which any given protein may be annotated. In GO representation, the parent-child relationship among terms implies that the child term is either a special case of the parent term or describes a process or component that is part of the parent process/component. In either case, there is a clear directional dependency. Specifically, a protein positively annotated to a child term is, by definition, also positively annotated to the parent term(s), but not vice versa. As a logical consequence, a protein that is negatively annotated to a parent term is also negatively annotated to the child term(s). A negative annotation indicates that a protein has been experimentally verified not to be involved in a particular function.

Researchers proposed a variety of methods for systematically predicting protein function considering its taxonomy structures at the same time. Here we list three representative approaches as following:

Markov Random Field Extension: A MRF model was extended to chain graphs in [11] to directly incorporate the structure of the Gene Ontology into the graphical representation for protein classification. The authors presented a method in which

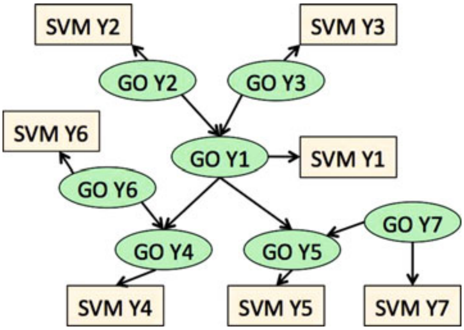


Fig. 21.19 A simple example of protein function identification considering the annotation taxonomy. Modified from Fig. 2 in [29]. SVM classifier is represented with *light red* node and GO terms are described with *green*. Here single SVM classifiers (with one SVM per function term) were combined through Bayesian networks to correct their predictions based on the hierarchical relationship between GO [14] terms

each protein is represented by a replicate of the Gene Ontology structure, effectively modeling each protein in its own annotation space. Belief propagation was used to make predictions at all ontology terms.

Ensemble Framework: Guan et al. [29] describe an ensemble framework based on SVMs that considers correlation between multiple function terms (see Fig. 21.19). A single SVM is used to predict a certain function for an unknown protein by integrating diverse datasets. In the context of the Gene Ontology hierarchy, single SVM classifiers are combined through Bayesian networks to correct their predictions based on the hierarchical relationship between GO terms in the GO directed acyclic graph. For each GO term, the method included all neighboring nodes in its Markov blanket to construct the Bayesian network. Shown in Fig. 21.19, $Y1$ is the GO node of interest in this example. Thus this Bayesian network was constructed with the local Markov blanket surrounding $Y1$.

Reconciliation Method: Similar to the above paper, Obozinski et al. [62] proposed to predict GO terms using an ensemble of discriminative classifiers. This paper focused on *reconciliation* methods for combining independent predictions to obtain a set of probabilistic predictions that are consistent with the topology of the ontology. Eleven distinct reconciliation methods were investigated: three heuristic methods; four variants of a Bayesian network; an extension of logistic regression to the structured case; and three novel projection methods including isotonic regression and two variants of a Kullback-Leibler projection method. The authors found that many apparently reasonable reconciliation methods yield reconciled probabilities with significantly lower precision than the original, unreconciled estimates. On the other hand, the isotonic regression method seems to be able to use the constraints from the GO network to its advantage, usually performing better than the underlying, unreconciled predictions.

Recently, in a special issue of *Genome Biology*, several research groups [64] used GO annotation as a benchmark to compare methods of protein function predictions with GO hierarchy structure being considered. Readers could refer to [64] for more discussion.

21.5 Related General Topics

All sub-problems covered in this chapter are instances of more general tasks like "link prediction", "entity labeling", "structural output learning" or "graph mining" in the machine learning, data mining, and social network analysis communities. Methods proposed in related research fields have great potentials to be used for protein-protein interaction prediction, protein function identification or domain-domain interaction detection in the near future. As the literature on these topics is vast, this section will briefly discuss just a few related studies as a guide.

21.5.1 Statistical Relational Learning (SRL)

As an area of growing interest in machine learning, statistical relational learning [25, 26] takes an object oriented approach to clearly distinguish between entities, relationships and their respective attributes in a probabilistic setting. Unlike most previous learning algorithms that assume all training examples are mutually independent, SRL methods try to capture complex relations among examples. A simple example of a relational system is a recommendation system: based on the attributes of two entities, i.e., of the user and the item, one wants to predict relationships like the preference (rating, willingness to purchase, ...) of this user for this item. One can exploit the known relationship attributes and the attributes of entities to predict unknown entity or relationship attributes [94]. This case is quite similar to protein-protein interaction prediction where we want to find the interaction preference of one protein to another. Various paradigms of SRL have been proposed in recent years, including probabilistic relational models, Bayesian logic programs, relational dependency networks, Markov logic networks, infinite relational model [43], infinite hidden relational model [94] and etc (surveyed in [25, 26, 59]). Several methods have software package available online, for instance, the open-source Alchemy system [44] provided a series of algorithms for statistical relational learning and probabilistic logic inference, based on the Markov logic representation [72]. It has been applied to problems in entity resolution, link prediction, information extraction and others [44].

21.5.2 Graph-Based Semi-Supervised Learning

Semi-supervised learning (SSL) [12] occupies the middle ground, between supervised learning (in which all training examples are labeled) and unsupervised learning (in which no label data is given). In application domains where unlabeled data are plentiful, such as bioinformatics, SSL got growing interests in recent years. One category of SSL algorithms consider dependencies between the labels of nearby examples on a constructed graph [9, 104] to perform joint inference. These models train to encourage nearby data points to have the same class labels, which is exactly protein function detection aims for. The graph-based SSL can obtain impressive performance using a very small amount of labeled data [12]. As we know from above, for a large number of protein functional categories, there exist very few annotated genes from experimental tests. Graph-based SSL might make better functional predictions for these classes. Mostafavi et al. [57] made some attemps in this direction.

21.5.3 Mining of Entity-Relation Graphs

In the data mining research community, relational or semi-structured data is naturally represented in a graph schema, where nodes denote entities and edges between nodes represent the relations between entities [22]. Such graphs are heterogeneous, since they include different types of nodes and different types of edges [56]. Many social networks could be described as entity-relation graphs. Using email system as an example, the graph inludes email-message, from-to-person, email-address and time entities which are inter-connected via relations derived from textual and structural information residing in a corporate database or a personal computer [56]. Similarly, protein interaction network could be converted to this schema easily where proteins, protein function annotations or domain compositions could be treated as different types of entities. Given an entity-relation graph, a popular question of interest is how to determine the nature of relationship between two entities that are not directly connected in the graph. The classical strategy [22] proposed in the literature performs random “lazy” graph walks on the entity-relation network to measure entity similarities. This strategy is closely related to graph-based SSL methods where “labels” (or “similarity”) from a start node propogate through edges in the graph, e.g., ccumulating evidence of relatedness over multiple connecting paths. The problem of “entity proximity” has connections to all three tasks we covered in this chapter. For instance, protein function prediction could be treated (“implicitly”) as a task of finding how similar an unknown protein is, to a known protein in terms of a specific functional category.

21.6 Summary

Biology relies on the concerted action of a number of biomolecules organized in networks, including proteins, small molecules, DNA and RNA. A key challenge is to understand the interactions among these molecules. The role of computational research on protein-protein interactions includes not only prediction, but also understanding the nature of the interactions and their binding residues on interaction interfaces. This chapter surveys recent efforts to predict interactions between proteins and between protein domains.

Predicting protein functions is one of the most important challenges of current computational biology research. A large number of computational techniques have been suggested for functional annotation using interaction networks; we have reviewed a few typical approaches in this chapter.

References

1. Abraham, I., Bartal, Y., Neimany, O. (2006). Advances in metric embedding theory. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing* (pp. 271–286). New York, NY, USA: ACM.
2. Airoldi, E. M., Blei, D. M., Fienberg, S. E., & Xing, E. P. (2008). Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, *9*, 1981–2014.
3. Arnau, V., Mars, S., & Marin, I. (2005). Iterative cluster analysis of protein interaction data. *Bioinformatics*, *21*(3), 364–378.
4. Bader, G. D., & Hogue, C. W. (2003). Analyzing yeast protein-protein interaction data obtained from different sources. *Nature Biotechnology*, *20*(10), 991–997.
5. Bader, J., Chaudhuri, A., Rothberg, J., & Chant, J. (2004). Gaining confidence in high-throughput protein interaction networks. *Nature Biotechnology*, *22*(1), 78–85.
6. Ben-Hur, A., & Noble, W. (2005). Kernel methods for predicting protein-protein interactions. *Bioinformatics (Proceedings of the Intelligent Systems for Molecular Biology Conference)*, *21*, i38–i46.
7. Ben-Hur, A., & Noble, W. (2006). Choosing negative examples for the prediction of protein-protein interactions. *BMC Bioinformatics*, *20*(Suppl. 1), S2.
8. Bleakley, K., Biau, G., & Vert, J. P. (2007). Supervised reconstruction of biological networks with local models. *Bioinformatics*, *23*(13), i57–i65. DOI 10.1093/bioinformatics/btm204. Retrieved from URL http://dx.doi.org/10.1093/bioinformatics/btm204
9. Blum, A. (2004). Semi-supervised learning using randomized mincuts. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*. Banff, Albert, Canada.
10. Braun, P., Tasan, M., Dreze, M., Barrios-Rodiles, M., Lemmens, I., Yu, H., Sahalie, J. M., Murray, R. R., Roncari, L., de Smet, A. S., Venkatesan, K., Rual, J. F., Vandenhaute, J., Cusick, M. E., Pawson, T., Hill, D. E., Tavernier, J., Wrana, J. L., Roth, F. P., & Vidal, M. (2009). An experimentally derived confidence score for binary protein-protein interactions. *Nature Methods*, *6*(1), 91–97. DOI 10.1038/nmeth.1281. Retrieved from URL http://dx.doi.org/10.1038/nmeth.1281.
11. Carroll, S., & Pavlovic, V. (2006). Protein classification using probabilistic chain graphs and the gene ontology structure. *Bioinformatics*, *22*, 1871–1878.
12. Chapelle, O., Schölkopf, B., & Zien, A. (eds.). (2006). *Semi-supervised learning. Adaptive computation and machine learning*. Cambridge: MIT.
13. Chia, J. M., & Kolatkar, P. R. (2004). Implications for domain fusion protein-protein interactions based on structural information. *BMC Bioinformatics*, *5*, 161.
14. Consortium, T. G. O. (2000). Gene ontology: tool for the unification of biology. *Nature Genetics*, *25*, 25–29.
15. Cusick, M. E., Yu, H., Smolyar, A., Venkatesan, K., Carvunis, A. R., Simonis, N., Rual, J. F., Borick, H., Braun, P., Dreze, M., Vandenhaute, J., Galli, M., Yazaki, J., Hill, D. E., Ecker, J. R., Roth, F. P., & Vidal, M. (2009). Literature-curated protein interaction datasets. *Nature Methods*, *6*(1), 39–46. DOI 10.1038/nmeth.1284. Retrieved from URL http://dx.doi.org/10.1038/nmeth.1284.
16. Davis, F. P., Barkan, D. T., Eswar, N., McKerrow, J. H., & Sali, A. (2007). Host pathogen protein interactions predicted by comparative modeling. *Protein Science*, *16*(12), 2585–2596. DOI 10.1110/ps.073228407. Retrieved from URL http://dx.doi.org/10.1110/ps.073228407.

17. Deng, M., Chen, T., & Sun, F. (2004). An integrated probabilistic model for functional prediction of proteins. *Journal of Computational Biology*, *11*(2–3), 463–475.
18. Deng, M., Mehta, S., Sun, F., & Chen, T. (2002). Inferring domain-domain interactions from protein-protein interactions. *Genome Research*, *12*(10), 1540–1548. Their method is actually an EM-based MLE.
19. Deng, M., Zhang, K., Mehta, S., Chen, T., & Sun, F. (2003). Prediction of protein function using protein-protein interaction data. *Journal of Computational Biology*, *10*(6), 947–960.
20. Dyer, M. D., Murali, T. M., & Sobral, B. W. (2007). Computational prediction of host-pathogen protein-protein interactions. *Bioinformatics*, *23*(13), i159–i166. DOI 10.1093/bioinformatics/btm208. Retrieved from URL http://dx.doi.org/10.1093/bioinformatics/btm208.
21. Espadaler, J., Romero-Isart, O., Jackson, R., & Oliva, B. (2005). Prediction of protein-protein interactions using distant conservation of sequence patterns and structure relationships. *Bioinformatics*, *21*(16), 3360–3368.
22. Faloutsos, C., Miller, G., & Tsourakakis, C. (2009). *Large graph-mining: Power tools and a practitioner's guide*. The 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Paris, June 28–July 2, 2009.
23. Gavin, A., Aloy, P., Grandi, P., et al. (2006). Proteome survey reveals modularity of the yeast cell machinery. *Nature*, *440*(7084), 631–636.
24. Gavin, A. C., Bosche, M., Krause, R., et al. (2002). Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, *415*(6868), 141–147. Retrieved from URL http://dx.doi.org/10.1038/415141a.
25. Getoor, L., & Diehl, C. (2005). Link mining: A survey. *SIGKDD Explorations*, *7*(2), 3–12.
26. Getoor, L., & Taskar, B. (2007). *Introduction to statistical relational learning*. Cambridge, MA: MIT.
27. Gomez, S., Noble, W., & Rzhetsky, A. (2003). Learning to predict protein-protein interactions from protein sequences. *Bioinformatics*, *19*(15), 1875–1881.
28. Gomez, S. M., Noble, W. S., & Rzhetsky, A. (2003). Learning to predict protein-protein interactions. *Bioinformatics*, *19*, 1875–1881.
29. Guan, Y., Myers, C. L., Hess, D. C., Barutcuoglu, Z., Caudy, A. A., & Troyanskaya, O. G. (2008). Predicting gene function in a hierarchical context with an ensemble of classifiers. *Genome Biology*, *9*(Suppl. 1), S3. DOI 10.1186/gb-2008-9-s1-s3. Retrieved from URL http://dx.doi.org/10.1186/gb-2008-9-s1-s3.
30. Han, D., Kim, H. S., Seo, J., & Jang, W. (2003). A domain combination based probabilistic framework for protein-protein interaction prediction. *Genome Information*, *14*, 250–259.
31. Hishigaki, H., Nakai, K., Ono, T., Tanigami, A., & Takagi, T. (2001). Assessment of prediction accuracy of protein function from protein–protein interaction data. *Yeast*, *18*(6), 523–531. DOI 10.1002/yea.706. Retrieved from URL http://dx.doi.org/10.1002/yea.706.
32. Ho, Y., Gruhler, A., Heilbut, A., Bader, G. D., Moore, L., Adams, S. L., et al. (2002). Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature*, *415*(6868), 180–183. Retrieved from URL http://dx.doi.org/10.1038/415180a.
33. Huang, C., Morcos, F., Kanaan, S. P., Wuchty, S., Chen, D. Z., & Izaguirre, J. A. (2007). Predicting protein-protein interactions from protein domains using a set cover approach. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **4**(1), 78–87. DOI 10.1109/TCBB.2007.1001. Retrieved from URL http://dx.doi.org/10.1109/TCBB.2007.1001.
34. Ingolfsson, H., & Yona, G. (2008). Protein domain prediction. *Methods in Molecular Biology*, *426*, 117–143.
35. Iqbal, M., Freitas, A. A., Johnson, C. G., & Vergassola, M. (2008). Message-passing algorithms for the prediction of protein domain interactions from protein-protein interaction data. *Bioinformatics*, *24*(18), 2064–2070. DOI 10.1093/bioinformatics/btn366.
36. Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., & Sakaki, Y. (2001). A comprehensive two-hybrid analysis to explore the yeast proteininteractome. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(8), 4569–4574. Retrieved from URL http://www.pnas.org/cgi/content/full/98/8/4569.

37. Jaimovich, A., Elidan, G., Margalit, H., & Friedman, N. (2006). Towards an integrated protein-protein interaction network: A relational markov network approach. *Journal of Computational Biology*, *13*(2), 145–164.
38. Jansen, R., & Gerstein, M. (2004). Analyzing protein function on a genomic scale: The importance of gold-standard positives and negatives for network prediction. *Current Opinion in Microbiology*, *7*, 535–545.
39. Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N. J., Chung, S., Emili, A., Snyder, M., Greenblatt, J. F., & Gerstein, M. (2003). A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, *302*, 449–453.
40. Jensen, L. J., Kuhn, M., Stark, M., Chaffron, S., Creevey, C., Muller, J., Doerks, T., Julien, P., Roth, A., Simonovic, M., Bork, P., & von Mering, C. (2009). String 8–a global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Research*, *37*(Database issue), D412–D416. DOI 10.1093/nar/gkn760. Retrieved from URL http://dx.doi.org/10.1093/nar/gkn760.
41. Karaoz, U., Murali, T., Letovsky, S., Zheng, Y., Ding, C., Cantor, C., & Kasif, S. (2004). Whole-genome annotation by using evidence integration in functional-linkage networks. *Proceedings of the National Academy of Sciences of the United States America*, *101*(9), 2888–2893.
42. Kato, T., Tsuda, K., & Asai, K. (2005). Selective integration of multiple biological data for supervised network inference. *Bioinformatics*, *21*(10), 2488–2495. DOI 10.1093/bioinformatics/bti339. Retrieved from URL http://dx.doi.org/10.1093/bioinformatics/bti339.
43. Kemp, C., Tenenbaum, J. B., Griffiths, T. L., Yamada, T., & Ueda, N. (2006). Learning systems of concepts with an infinite relational model. In A. Cohn (Ed.), *Proceedings of the 21st national conference on artificial intelligence (AAAI' 06)* (Vol. 1, pp. 381–388). AAAI Press.
44. Kok, S., Sumner, M., Richardson, M., Singla, P., Poon, H., Lowd, D., & Domingos, P. (2007). *The alchemy system for statistical relational ai*. Technical report of Department of Computer Science and Engineering, Washington, USA: University of Washington.
45. Kondor, R. I., & Lafferty, J. D. (2002). Diffusion kernels on graphs and other discrete input spaces. In *ICML '02: Proceedings of the nineteenth international conference on machine learning* (pp. 315–322). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
46. Lanckriet, G., Deng, M., Cristianini, N., Jordan, M., & Noble, W. (2004). Kernel-based data fusion and its application to protein function prediction in yeast. *The Ninth Pacific Symposium on Biocomputing (PSB 2004)*, 300–311.
47. Lee, I., Date, S. V., Adai, A. T., & Marcotte, E. M. (2004). A probabilistic functional network of yeast genes. *Science*, *306*, 1555–1558.
48. Lee, I., Lehner, B., Crombie, C., Wong, W., Fraser, A. G., & Marcotte, E. M. (2008). A single gene network accurately predicts phenotypic effects of gene perturbation in caenorhabditis elegans. *Nature Genetics*, *40*(2), 181–188. DOI 10.1038/ng.2007.70. Retrieved from URL http://dx.doi.org/10.1038/ng.2007.70.
49. Leone, M., & Pagnani, A. (2004). Predicting protein functions with message passing algorithms. *Bioinformatics*, *21*(2), 239–247.
50. Letovsky, S., & Kasif, S. (2003). Predicting protein function from protein/protein interaction data: A probabilistic approach. *Bioinformatics*, *19*(Suppl. 1), I197–I204.
51. Lin, N., Wu, B., Jansen, R., Gerstein, M., & Zhao, H. (2004). Information assessment on predicting protein-protein interactions. *BMC Bioinformatics*, *5*, 154.
52. Marcotte, E. M., Pellegrini, M., Ng, H. L., Rice, D. W., Yeates, T. O., & Eisenberg, D. (1999). Detecting protein function and protein-protein interactions from genome sequences. *Science*, *285*, 751–753.
53. Martin, S., Roe, D., & Faulon, J. L. (2005). Predicting protein-protein interactions using signature products. *Bioinformatics*, *21*(2), 218–226.
54. von Mering, C., Jensen, L., Snel, B., Hooper, S., Krupp, M., Foglierini, M., Jouffre, N., Huynen, M., & Bork, P. (2005). STRING: Known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic Acids Research*, *33*, D433–D437.

55. von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S. G., Fields, S., & Bork, P. (2002). Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, *417*(6887), 399–403.
56. Minkov, E. (2008). *Adaptive graph walk based similarity measures in entity-relation graphs*. Ph.D. thesis, Carnegie Mellon University.
57. Mostafavi, S., Ray, D., Warde-Farley, D., Grouios, C., & Morris, Q. (2008). Genemania: A real-time multiple association network integration algorithm for predicting gene function. *Genome Biology*, *9*(Suppl. 1), S4. DOI 10.1186/gb-2008-9-s1-s4. Retrieved from URL http://dx.doi.org/10.1186/gb-2008-9-s1-s4.
58. Nabieva, E., Jim, K., Agarwal, A., Chazelle, B., & Singh, M. (2005). Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, *21*(S1), i302–i310.
59. Neville, J., Rattigan, M., & Jensen, D. (2003). Statistical relational learning: Four claims and a survey. In: Getoor, L., & Jensen, D. (Eds.) *The workshop on learning statistical models from relational data, 18th international joint conference on artificial intelligence*. Mexico: Acapulco.
60. Nguyen, T. P., & Ho, T. B. (2008). An integrative domain-based approach to predicting protein-protein interactions. *Journal of Bioinformatics and Computational Biology*, *6*(6), 1115–1132.
61. Nye, T. M. W., Berzuini, C., Gilks, W. R., Babu, M. M., & Teichmann, S. A. (2005). Statistical analysis of domains in interacting protein pairs. *Bioinformatics*, *21*(7), 993–1001. DOI 10.1093/bioinformatics/bti086. Retrieved from URL http://dx.doi.org/10.1093/bioinformatics/bti086.
62. Obozinski, G., Lanckriet, G., Grant, C., Jordan, M. I., & Noble, W. S. (2008). Consistent probabilistic outputs for protein function prediction. *Genome Biology*, *9*(Suppl. 1), S6. DOI 10.1186/gb-2008-9-s1-s6. Retrieved from URL http://dx.doi.org/10.1186/gb-2008-9-s1-s6.
63. Pagel, P., Strack, N., Oesterheld, M., Stmpflen, V., & Frishman, D. (2007). Computational prediction of domain interactions. *Methods of Molecular Biology*, *396*, 3–15.
64. Peña-Castillo, L., Tasan, M., Myers, C. L., Lee, H., Joshi, T., Zhang, C., Guan, Y., Leone, M., Pagnani, A., Kim, W. K., Krumpelman, C., Tian, W., Obozinski, G., Qi, Y., Mostafavi, S., Lin, G. N., Berriz, G. F., Gibbons, F. D., Lanckriet, G., Qiu, J., Grant, C., Barutcuoglu, Z., Hill, D. P., Warde-Farley, D., Grouios, C., Ray, D., Blake, J. A., Deng, M., Jordan, M. I., Noble, W. S., Morris, Q., Klein-Seetharaman, J., Bar-Joseph, Z., Chen, T., Sun, F., Troyanskaya, O. G., Marcotte, E. M., Xu, D., Hughes, T. R., & Roth, F. P. (2008). A critical assessment of mus musculus gene function prediction using integrated genomic evidence. *Genome Biology*, *9*(Suppl. 1), S2. DOI 10.1186/gb-2008-9-s1-s2. Retrieved from URL http://dx.doi.org/10.1186/gb-2008-9-s1-s2.
65. Pellegrini, M., Marcotte, E. M., Thompson, M. J., Eisenberg, D., & Yeates, T. O. (1999). Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proceedings of the National Academy of Sciences of the United States of America*, *96*(8), 4285–4288.
66. Qi, Y., Bar-Joseph, Z., & Klein-Seetharaman, J. (2006). Evaluation of different biological data and computational classification methods for use in protein interaction prediction. *PROTEINS: Structure, Function, and Bioinformatics*, *63*(3), 490–500.
67. Qi, Y., Klein-Seetharaman, J., & Bar-Joseph, Z. (2005). Random forest similarity for protein-protein interaction prediction from multiple sources. *Pacific Symposium on Biocomputing*, *10*, 531–542.
68. Qi, Y., Klein-Seetharaman, J., & Bar-Joseph, Z. (2005). Random forest similarity for protein-protein interaction prediction from multiple sources. In *Proceedings of the Pacific Symposium*, Hawaii, USA, 4–8 January 2005, pp. 531–542.
69. Qiu, J., & Noble, W. S. (2008). Predicting co-complexed protein pairs from heterogeneous data. *PLoS Computational Biology*, *4*(4), e1000,054. DOI 10.1371/journal.pcbi.1000054. Retrieved from URL http://dx.doi.org/10.1371/journal.pcbi.1000054.

70. Ramani, A. K., Bunescu, R. C., Mooney, R. J., & Marcotte, E. M. (2005). Consolidating the set of known human protein-protein interactions in preparation for large-scale mapping of the human interactome. *Genome Biology*, *6*(5), R40. Article.
71. Rhodes, D. R., Tomlins, S. A., Varambally, S., Mahavisno, V., Barrette, T., Kalyana-Sundaram, S., Ghosh, D., Pandey, A., & Chinnaiyan, A. M. (2005). Probabilistic model of the human protein-protein interaction network. *Nature Biotechnology*, *8*, 951–959. Article.
72. Richardson, M., & Domingos, P. (2006). Markov logic networks. *Machine Learning*, *62*, 107–136.
73. Riley, R., Lee, C., Sabatti, C., & Eisenberg, D. (2005). Inferring protein domain interactions from databases of interacting proteins. *Genome Biology*, *6*(10), R89. DOI 10.1186/gb-2005-6-10-r89. Retrieved from URL http://dx.doi.org/10.1186/gb-2005-6-10-r89.
74. Ritchie, D. W. (2008). Recent progress and future directions in protein-protein docking. *Current Protein and Peptied Science*, *9*(1), 1–15.
75. Rual, J. F., Venkatesan, K., et al. (2005). Towards a proteome-scale map of the human protein-protein interaction network. *Nature*, *437*(7062), 1173–1178. 1476–4687 (Electronic) Journal Article.
76. Samanta, M., & Liang, S. (2003). Predicting protein functions from redundancies in large-scale protein interaction networks. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(22), 12,579–12,583.
77. Schwikowski, B., Uetz, P., & Fields, S. (2000). A network of protein-protein interactions in yeast. *Nature Biotechnol*, *18*(12), 1257–1261. DOI 10.1038/82360. Retrieved from URL http://dx.doi.org/10.1038/82360
78. Scott, M. S., & Barton, G. J. (2007). Probabilistic prediction and ranking of human protein-protein interactions. *BMC Bioinformatics*, *8*, 239. 1471–2105 (Electronic) Comparative Study Journal Article Research Support, Non-U.S. Gov't
79. Scudder, H. (1965). Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, *11*(3), 363–371.
80. Sharan, R., Ulitsky, I., & Shamir, R. (2007). Network-based prediction of protein function. *Molecular Systems Biology*, *3*, 88. 1744–4292 (Electronic) Journal Article Research Support, Non-U.S. Gov't Review
81. Shoemaker, B. A., & Panchenko, A. R. (2007). Deciphering protein-protein interactions. part i. experimental techniques and databases. *PLoS Computational Biology*, *3*(3), e42. 1553–7358 (Electronic) Journal Article Research Support, N.I.H., Intramural Review.
82. Shoemaker, B. A., & Panchenko, A. R. (2007). Deciphering protein-protein interactions. part ii. computational methods to predict protein and domain interaction partners. *PLoS Computational Biology*, *3*(4), e43. 1553–7358 (Electronic) Journal Article Research Support, N.I.H., Intramural Review.
83. Smola, A., & Kondor, R. (2003). Kernels and regularization on graphs. In B. Schölkopf & M. Warmuth (Eds.), *Proceedings of the annual conference on computational learning theory and kernel workshop, lecture notes in computer science*. Germany: Springer-Verlag.
84. Sontag, D., Singh, R., & Berger, B. (2007). Probabilistic modeling of systematic errors in two-hybrid experiments. *The Twelfth Pacific Symposium on Biocomputing (PSB 2007)*, 445–457.
85. Sprinzak, E., & Margalit., H. (2001). Correlated sequence-signatures as markers of protein-protein interaction. *Journal of Molecular Biology*, *311*, 681-692. Use mutual information (average) of two sequence signatures in the interacting protein pairs as signature interact probability; InterPro => sequence signature of protein.
86. Stelzl, U., Worm, U., Lalowski, M., et al. (2005). A human protein-protein interaction network: A resource for annotating the proteome. *Cell*, *122*(6), 957–968. 0092–8674 (Print) Journal Article.
87. Tastan, O., Qi, Y., Carbonell, J., & Klein-Seetharaman, J. (2009). Prediction of interactions between hiv-1 and human proteins by information integration. *The fourteenth Pacific Symposium on Biocomputing (PSB 2009)*, pp. 516–527.
88. Teichmann, S. A. (2002). Principles of protein-protein interactions. *Bioinformatics*, *18*(Suppl. 2), S249.

89. Tsuda, K., & Noble, W. (2004). Learning kernels from biological networks by maximizing entropy. *Bioinformatics*, *20*(Suppl. 1), I326–I333.
90. Vazquez, A., Flammini, A., Maritan, A., & Vespignani, A. (2003). Global protein function prediction from protein-protein interaction networks. *Nature Biotechnology*, *21*, 697–700.
91. Vert, J. P., Qiu, J., & Noble, W. S. (2007). A new pairwise kernel for biological network inference with support vector machines. *BMC Bioinformatics*, *8*(Suppl. 10), S8
92. Wang, H., Segal, E., Ben-Hur, A., Li, Q., Vidal, M., & Koller, D. (2007). InSite: A computational method for identifying protein-protein interaction binding sites on a proteome-wide scale. *Genome Biology*, *8*(9), R192.1–R192.18.
93. Wu, Y., & Lonardi, S. (2008). A linear-time algorithm for predicting functional annotations from ppi networks. *Journal of Bioinformatics and Computational Biology*, *6*(6), 1049–1065.
94. Xu, Z., Tresp, V., Yu, K., & Kriegel, H. P. (2006). Infinite hidden relational models. In *UAI '06, Proceedings of the 22nd Conference in Uncertainty in Artificial Intelligence*, July 13–16 2006, Cambridge, MA, USA.
95. Yamanishi, Y., Araki, M., Gutteridge, A., Honda, W., & Kanehisa, M. (2008). Prediction of drug-target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics*, *24*(13), i232–i240. DOI 10.1093/bioinformatics/btn162. Retrieved from URL http://dx.doi.org/10.1093/bioinformatics/btn162.
96. Yamanishi, Y., Vert, J. P., & Kanehisa, M. (2004). Protein network inference from multiple genomic data: A supervised approach. *Bioinformatics*, *20*(Suppl. 1), i363–i370. DOI 10.1093/bioinformatics/bth910. Retrieved from URL http://dx.doi.org/10.1093/bioinformatics/bth910.
97. Yanay, O., Marco, P., & Burkard, R. (2005). Tutorial: Function prediction – from high throughput to individual proteins. *The Tenth Pacific Symposium on Biocomputing (PSB 2005)*.
98. Yip, K.Y., & Gerstein, M. (2009). Training set expansion: An approach to improving the reconstruction of biological networks from limited and uneven reliable interactions. *Bioinformatics*, *25*(2), 243–250. DOI 10.1093/bioinformatics/btn602. Retrieved from URL http://dx.doi.org/10.1093/bioinformatics/btn602.
99. Yu, H., Braun, P., Yildirim, M.A., Lemmens, I., Venkatesan, K., Sahalie, J., Hirozane-Kishikawa, T., Gebreab, F., Li, N., Simonis, N., Hao, T., Rual, J. F., Dricot, A., Vazquez, A., Murray, R. R., Simon, C., Tardivo, L., Tam, S., Svrzikapa, N., Fan, C., de Smet, A. S., Motyl, A., Hudson, M. E., Park, J., Xin, X., Cusick, M. E., Moore, T., Boone, C., Snyder, M., Roth, F. P., Barabsi, A. L., Tavernier, J., Hill, D. E., & Vidal, M. (2008). High-quality binary protein interaction map of the yeast interactome network. *Science*, *322*(5898), 104–110. DOI 10.1126/science.1158684. Retrieved from URL http://dx.doi.org/10.1126/science.1158684
100. Yu, J., & Finley, R. L. (2009). Combining multiple positive training sets to generate confidence scores for protein-protein interactions. *Bioinformatics*, *25*(1), 105–111. DOI 10.1093/bioinformatics/btn597. Retrieved from URL http://dx.doi.org/10.1093/bioinformatics/btn597.
101. Zhang, L., Wong, S., King, O., & Roth, F. (2004). Predicting co-complexed protein pairs using genomic and proteomic data integration. *BMC Bioinformatics*, *5*, 38.
102. Zhang, L. V., Wong, S., King, O., & Roth, F. (2004). Predicting co-complexed protein pairs using genomic and proteomic data integration. *BMC Bioinformatics*, *5*(1), 38–53.
103. Zhou, H. X., & Qin, S. (2007). Interaction-site prediction for protein complexes: A critical assessment. *Bioinformatics*, *23*(17), 2203–2209. DOI 10.1093/bioinformatics/btm323. Retrieved from URL http://bioinformatics.oxfordjournals.org/cgi/content/abstract/23/17/2203.
104. Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In: T. Fawcett & N. Mishra (Eds.), The Twentieth International Conference on Machine Learning (ICML-2003), Washington, DC. Menlo Park, California: AAAI Press. ISBN 978-1-57735-189-4

Chapter 22
Reverse Engineering of Gene Regulation Networks with an Application to the DREAM4 *in silico* Network Challenge

Hyonho Chun, Jia Kang, Xianghua Zhang, Minghua Deng, Haisu Ma, and Hongyu Zhao

Abstract Despite much research, reverse engineering of gene regulation remains a challenging task due to a large number of genes involved and complex relationships among them. In this chapter, we review statistical methods for inferring gene regulation networks, specifically focusing on the methods for analyzing gene expression data. We then present a new reverse engineering method in order to efficiently utilize datasets from various perturbation experiments as well as to integrate these multiple sources of information. We apply our approach to the DREAM *in silico* network challenge to demonstrate its performance.

22.1 Introduction

Inferring gene regulatory networks (GRNs) has been a vigorous research area as a result of the increasing availability of genome wide gene expression data [13, 20, 23, 42]. However, reverse engineering remains a challenging task, because a large number of genes are involved in GRNs where the complexity of the network inference problem increases super-exponentially [11] as a function of the number

H. Chun
Department of Epidemiology and Public Health, Yale University

J. Kang
Program in Translational Informatics, Yale University

X. Zhang
Department of Electronic Science and Technology, University of Science and Technology of China

M. Deng
School of Mathematical Sciences and Center for Theoretical Biology, Peking University

H. Ma
Program in Computational Biology, Yale University

H. Zhao (✉)
Department of Genetics, Yale University
e-mail: hongyu.zhao@yale.edu

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_22,
© Springer-Verlag Berlin Heidelberg 2011

of genes involved. To promote further research, an unbiased assessment of reverse engineering methods such as DREAM [26, 43, 44] has been advocated to compare the performance of different methods recently.

In this chapter, we first review statistical methods for inferring GRNs, specifically focusing on the methods for analyzing gene expression data. We also describe typical perturbation experiments for generating gene expression data, where only partial information on a GRN can be obtained from each dataset. We then present a new reverse engineering method in order to efficiently extract information that is specific to each dataset as well as to integrate these multiple sources of information. We apply our approach to the DREAM4 *in silico* network challenge to demonstrate the performance of our method, and a brief conclusion will follow.

22.2 Statistical Methods for the Network Inference from Gene Expression Data

In this subsection, we review statistical methods that have been developed to infer GRNs, specifically focusing on the methods for analyzing gene expression data. In this setting, gene expressions are measured after perturbing a system of genes in various ways in order to reveal the regulatory structure in the system. There are generally two types of perturbation datasets for the GRN inference. In the first type, a number of perturbations are studied but only one observation is made for each perturbation after genes reach steady states. For the second type, gene expression profiles are recorded along a number of time points. In our following discussion, we will refer to these two types as steady state and time course, respectively.

Intuitively, genes in a pathway will likely change together in response to a perturbation, and one method for network reconstruction is to examine the association of expression profiles between each pair of genes. A network can be built based on pairwise dependencies [7, 8], called Relevance Network (RN). The association between genes can be measured by Pearson correlation coefficient or its various transformations, or mutual information measures; then those edges that are more associated than a pre-specified threshold or empirically determined threshold are retained to construct the network. Although appealing, one major limitation of this approach is that pairwise dependencies may be due to either direct regulatory relationship or indirect relationship through other genes. Therefore, the networks inferred from pairwise association may contain many false positive edges. One way to reduce false positives is to study the association between two genes in the presence of one or more other genes, where examples include ARCANE [27] and conditional correlation coefficients [33, 49].

Because genes in a system are likely to interact together, it may not be ideal to consider two, or at most three genes at a time, as in the RN approach. To model all genes together, Gaussian Graphical models (GGM) have been introduced. As GGMs assume that gene expression measurements are sampled from a multivariate normal distribution with covariance matrix Σ, the modeling of genes only involves estimating a mean vector and a covariance matrix. One attractive feature of GGM is that the inverse of the covariance matrix, or the precision matrix (Σ^{-1}), represents the conditional correlation of any pair of genes conditional on all other genes. Thus GGM formulation will allow us to distinguish gene pairs that directly interact from those that indirectly interact through other genes.

However, when the number of experiments is not much larger or smaller than the number of genes, which is often the case for gene expression data, the estimated covariance matrix $\hat{\Sigma}$ can be singular and it is difficult to estimate a stable precision matrix. As a remedy, several attempts are made by utilizing conditional mean model representation of GGM with sparsity assumption imposed. Specifically, a characterization of $(Y_1, \ldots, Y_G)$ being the multivariate Gaussian distribution is provided by Fisk [17], and the main characteristic is that the conditional distribution of Y_i given the remaining $Y_1, \ldots, Y_{i-1}, Y_{i+1}, \ldots, Y_G$ depends on conditioning vectors only through the conditional mean $\beta_{i0} + \sum_{j=1, j\neq i}^{G} \beta_{ij} m_j$, for all i, where G is the total number of genes, m_j is the mean expression of gene j, β_{ij}s are the coefficients of the jth gene on the ith gene, and β_{i0} is the constant term for the ith gene. We then represent the gene expression measurement of the ith gene in following linear regression model:

$$Y_{ik} = \beta_{i0} + \sum_{j=1, j\neq i}^{G} \beta_{ij} Y_{jk} + \epsilon_{ik},$$

where Y_{ik} is the expression measurement of gene i at the kth experiment, and ϵ_{ik} is a Gaussian noise term. This representation enables the integration of many regularization techniques (e.g. LASSO) to avoid singularity problem caused by small number of experiments.

In network analysis, the sparsity assumption, e.g. there are a small number of edges for each node, is often made because biological networks are usually assumed to be sparse. The most popular method of incorporating this assumption is by using LASSO [40], and Meinshausen and Buhlmann [28] performed the regularized regression method LASSO in the following form:

$$\hat{\beta}_i^{\lambda} = \text{argmin}_{\beta_i} \left\{ \left\| Y_i - \beta_{0i} - \sum_{j=1, j\neq i}^{G} \beta_{ij} Y_j \right\|_2^2 + \lambda \sum_{j=1, j\neq i}^{G} |\beta_{ij}| \right\},$$

where $\hat{\beta}_i^{\lambda}$ is the regression coefficient vector for gene i, and λ is the penalty term that reflects the balance between model fit and model complexity. This procedure often leads to a model with only a few non-zero regression coefficients. However, the resulting network structure may not be symmetric, because each gene is fitted separately, which will lead to difficulties in network interpretation. Later, Peng et al. [30] achieved network symmetry by reformulating the coefficients β_{ij}s into partial correlations and then regularize these partial correlations while fitting all genes simultaneously. Some other approaches under the regression setting include iterative greedy algorithms and combinatorial algorithms [1, 2]. In addition, a threshold gradient decent method is proposed to estimate the precision matrix [25]. Furthermore, empirical Bayes or shrinkage approaches can be used to infer large scale GRNs [37, 38], and the sparsity assumption can be accomplished by prior specification under the Bayesian setting [14, 34].

Bayesian networks (BNs) were proposed very early to infer gene networks based on gene expression data [18,22]. In BNs, gene regulations are modeled as a directed acyclic graph (DAG), and the joint distribution of all the genes can be formulated as a product of conditional probabilities. When inferring the structure of an underlying DAG, a term reflecting the balance between model fit and model complexity is incorporated to prevent overfitting. Although gene network inference by using BN approaches is an active research field [15], a number of limitations remain to be addressed [45]. First, there are cases where the DAGs cannot be differentiated purely from the observed datasets (e.g. DAGs forming an equivalence class). Second, continuous observations need to be discretized for BN analysis in most cases. Third, BNs do not allow feedback loops, a common phenomenon in biological networks. To allow feedback loops, dynamic Bayesian networks (DBNs) have been proposed [31, 51]. But DBNs typically assume time homogeneous transition model, resulting oversimplified dynamics of the biological processes.

Time course gene expression data can also be utilized to infer regulatory relationships. One approach is to use linear regression model in the light of ordinary differential equation (ODE):

$$\frac{Y_i(t+\Delta t) - Y_i(t)}{\Delta t} = \sum_{j=1, j\neq i}^{G} \beta_{ij} Y_j(t) - d_i Y_i(t) + \epsilon_i(t+\Delta t),$$

where the dependent variable is the change of expression level between two observations, and d_i represents the decay rate of the ith gene. Bansal et al. [3] developed the TSNI algorithm, where the observed time course data are first smoothed by using splines and then SVD is applied to the gene expression matrix to reduce the dimensionality of the predictors. We remark that when both steady state and time course data are available, a single regression model can be used [5] by representing steady state equation as

$$0 = \sum_{j=1, j\neq i}^{G} \beta_{ij} Y_j - d_i Y_i.$$

To impose sparsity to the biological network, Gardner et al. [19] select k regulators for each gene that fit expression data with the smallest error. In addition, forward model selection using AIC is also widely adopted [10]. Recently, regularization methods have been utilized under the ODE framework [12].

Another approach utilizes nonlinear regression model [47]:

$$\frac{Y_i(t+\Delta t) - Y_i(t)}{\Delta t} = \left[\alpha_i \prod_j Y_j^{w_{ji}}(t) \prod_k Y_k^{w_{ki}}(t-\Delta t) \prod_l Y_l^{w_{li}}(t-2\Delta t) - d_i Y_i(t)\right] + \epsilon_i(t+\Delta t),$$

where w_{ji}, w_{ki} and w_{li} are the coefficients to quantify the effects of corresponding gene, and d_i is the mRNA decay rate. The time-delayed response incorporates gene expression levels at time $t, t - \Delta t$ and $t - 2\Delta t$, and the joint effects from multiple genes are modeled through a multiplicative function. This model can be considered as a variant of S-system [35,36] . To overcome the difficulty of parameter estimation, the authors use a genetic algorithm coupled with Bayesian Information Criterion (BIC)[45] and only consider a limited set of genes when this model is applied. Additionally, the problem of estimating nonlinear ODE models for gene regulation network using the generalized profiling method for functional data analysis [32] is also studied by Cao and Zhao [9].

Vector autoregressive (VAR) network model is used to infer gene network from time course gene expression datasets, where the model is given by:

$$Y_i(t + \Delta t) = \beta_{i0} + \beta_{ii}(t)Y_i(t) + \sum_{j=1, j \neq i}^{G} \beta_{ij} Y_j(t) + \epsilon_i(t + \Delta t).$$

The nonzero coefficients of the model represent directed causal influence [21]. Opgen-Rhein and Strimmer [29] use shrinkage method on VAR network model to accommodate the singularity problem due to a large number of genes with a small number of experimental conditions. Alternatively, elastic net method is proposed for this purpose [39].

The performances of RNs, GGMs and BNs have been compared in the literature [41,48]. Both BNs and GGMs have better performances than RNs. And, for a small system, BNs outperform GGMs and structural perturbations are more informative than dynamic perturbations (e.g. time course data) [45].

22.3 Reverse Engineering of Gene Expression Datasets from Various Perturbation Experiments

In the previous section, we reviewed statistical methods that had been developed in recent years to reconstruct gene regulatory networks, focusing on methods designed for analyzing gene expression data. However, any single method cannot be optimal to study datasets from various types of perturbation experiments. Therefore, in this section, we first describe typical perturbation experiments for generating gene expression data and then propose a method that is aiming to infer GRNs by extracting experiment-specific information across datasets from various experiments and then integrating these information in a systematic way.

The steady state mRNA level of unperturbed networks is referred to as "wild type". Various perturbation experiments can be performed to induce deviations in the gene expression levels from wild types. The first is to reduce the transcription rate of a single gene or down to zero while keeping all the other genes constant, which are referred to as knock down or knock out experiments, respectively.

Independent knock down/out experiment can be performed for every gene in the network. The second is to slightly increase or decrease the basal activation of all genes in the network simultaneously by random magnitudes, which are called the multi-factorial experiments. These multi-factorial perturbations can be repeated as many times as the number of genes in the network. Each set of gene expressions can be measured either at the system's new steady state (steady state dataset), or across several times points (time course dataset). From now on, we assume that datasets from both steady states and time course experiments are available.

In order to use RN, GGM and BN approaches on the knock down/out data, different perturbation experiments need to be treated interchangeably. However, since the underlying biological states are clearly different [45], this may lead to loss of valuable information. For example, knock down/out data contain the information on which gene is perturbed in each experiment, but this information is not directly utilized in these methods. Therefore, we propose a method that may analyze knock down/out datasets more efficiently. We remark that we did not pursue BNs in the subsequent analysis, since mRNA measurements are hard to be optimally discretized, which is a major limitation of BNs [45].

Since knock down/out datasets reflect series of subsystems by keeping the values of most parameters constant and varying only one of them, they could be the most valuable sources of inferring network structures [50]. Without considering noise, if the expression level of gene B changes significantly as a result of perturbing gene A, this could imply a direct path from A to B. However, in the presence of noise, the relationship between gene A and gene B could be a result of either true signal or noise, and hence a crucial step in inferring the network structure is to extract reliable regulatory signals from the background noise.

The mRNA levels for genes in the wild type and knock down/out experiments contain only a single data point. In order to facilitate the statistical modeling of background noise, it is necessary to effectively combine various datasets across experiments, and we achieve data integration by assuming the noise of gene i's expression follows the same normal distribution with mean 0 and standard deviation σ_i across different datasets. This is a natural assumption to make because the data were generated from the same set of dynamic models. We remark that we assume the noise is from a normal distribution for the model simplicity.

To detect gene expression changes, it is important to accurately estimate the wild type (or base line) expression and the background noise. Although using knock down/out data to estimate these variables (e.g. the error model of Yip [50]) may appear to be an appealing strategy, we recognize that parameter estimation using knock down/out data alone often heavily relies on the assumption of signal sparsity, which is not always met in practice. In addition, the signal in the time course data, or the time trend, has the nice property of being continuous. Once the smooth time trend is estimated, we can compute the residuals by subtracting the estimated time trend from the observed expression levels in the time course. The residuals can then be used to model the background noise. Note that the procedures outlined above do not rely upon the sparsity assumption, and therefore, the time course data is chosen to estimate the background noise.

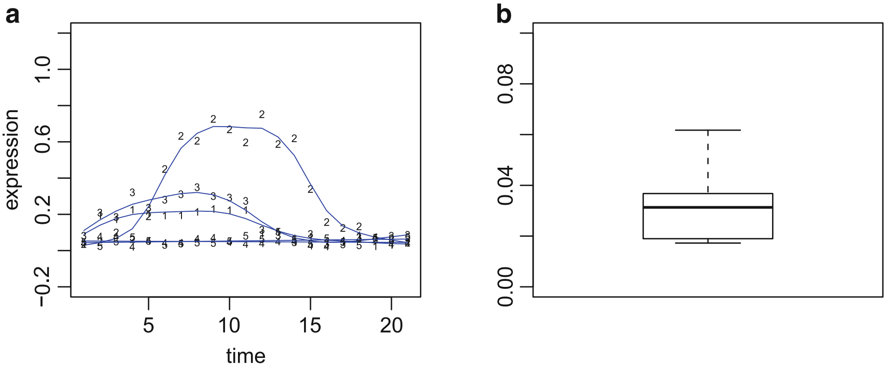


Fig. 22.1 Variance estimation by using time course datasets. (**a**) Seperate time trend estimates across replicates by using smoothing spline method. (**b**) Box plot of the standard deviation estimates across replicates

We adopt the smoothing spline method to estimate the time trend, and the degree of smoothness, or the smoothing parameter, is determined using the generalized maximum likelihood (GML) method [46]. We separately estimate the time trend and the standard deviation for each replicate, and then take their median value as the final estimate of the standard deviation (Fig. 22.1). The wild type expression level for each gene is estimated by simply averaging the values that correspond to wild type measurements from both steady state and time course datasets.

With the estimated wild type ($\hat{m}_i$) and standard deviation of noise ($\hat{\sigma}_i$), we can effectively utilize knock down/out data to generate scores that should separate signals from background noise, by computing the p-value associated with the regulation of gene i to gene j as $2\left(1 - \Phi\left(|Y_{ij} - \hat{m}_i|/\hat{\sigma}_i\right)\right)$, where Φ is the cumulative distribution function of standard normal random variable, and Y_{ij} is the expression of gene i when gene j is knocked down/out. Although the scores generated from knock down/out experiments are quite informative for learning network structure, a couple of problems may hamper an accurate inference. First, there are situations where knock down/out experiments cannot convey useful information, resulting in false negatives. For example, if a gene (e.g. gene A) has a low wild type expression level and is positively regulated by another gene (e.g. gene B), knocking down/out gene B will only impose a very limited impact on gene A's expression. Similarly, if gene A's wild type expression is already saturated at its maximum, knocking down/out its negative regulators (e.g. gene B) will not induce dramatic changes on gene A's expression either. Second, regulations detected from the knock down/out experiments inherently may contain many indirect relationships, and including these indirect edges will clearly elevate the number of false positives.

Because multi-factorial data contain changes of expression levels resulting from perturbations both in the up and down directions, we utilize multi-factorial perturbation data to complement the estimates obtained from the knock down/out experiments (Fig. 22.2). Specifically, we use pair-wise correlation to quantify the

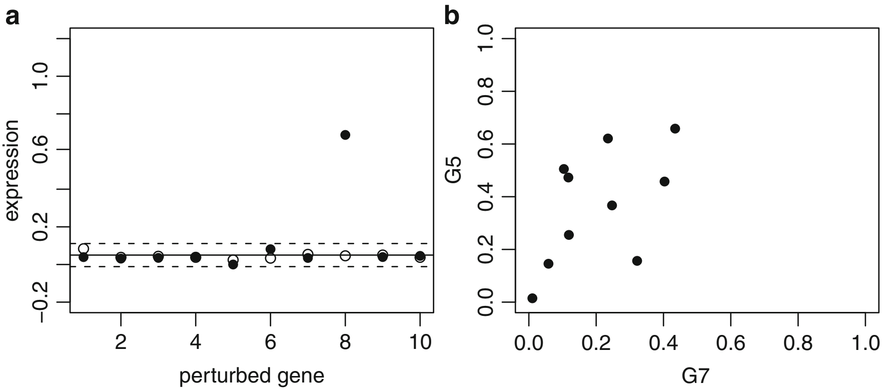


Fig. 22.2 Limitation of knock out and knock down experiments. (**a**) *Solid dots* represent knock out experiments and circles represent knock down experiments of gene 5 of the 5th network of DREAM4 *in silico* network challenge. *Solid line* represent the estimated wild type expression and *dotted lines* 99% confidence intervals. Wild type is low and gene 5 does not change in gene 7 knock out experiment. (**b**) *Scatter plot* of gene expression of gene 5 and gene 7 from multi-factorial experiments. A positive correlation is observed as the expression level of gene 7 increases

regulatory strength between a pair of genes on the basis that these genes tend to respond similarly to different perturbations as in the RN approach [7, 8]. Then, we use p-values of the pair-wise correlations as our third score for the network inference. This approach should alleviate the aforementioned false negative problem; however, it is worth noting that the direction of regulation cannot be inferred from this dataset alone, since all of the genes in the network are perturbed simultaneously in each multi-factorial experiment.

On the other hand, to effectively address the false positive issue, a conditional analysis [33, 49] is required to differentiate direct from indirect regulations among genes as we mentioned in previous section. One convenient way would be computing the conditional correlation coefficients between genes conditional on all other genes in the multi-factorial datasets via GGMs. However, when multi-factorial datasets contain too few experiments, conditional correlations cannot be accurately estimated without further modifications such as by employing regularization methods [28]. Although imposing regularization appears to be an appealing strategy, we found that it did not drastically improve the performance when the sample size (the number of experimental conditions) is small and the covariates (gene expressions) are highly correlated. Therefore, we utilize time course datasets for this purpose via ODE. Since time course datasets have replicates with many time points, solving ODE is not too difficult. Denoting $Y_i^r(t)$ as the expression level of gene i at time point t of the rth replicate, we can write down the ODE model in the following way:

$$\frac{Y_i^r(t+\Delta t) - Y_i^r(t)}{\Delta t} = \beta_{i0} + \sum_{1}^{G} \beta_{ik} Y_k^r(t) + \epsilon_i^r(t+\Delta t),$$

where G represents the total number of genes in the network. To estimate the parameters of the ODE model, we use the ordinary least squares approach. We use the p-values for the hypothesis of $\beta_{ij} = 0$ given β_{ik}s where $k \neq j$, as our score to infer the direct regulation of the gene j on gene i.

So far, we have generated four scores using the knockout, knockdown, multifactorial, and time course datasets. We combine these scores by Fisher's method [16], where the test statistics is given by $\chi_{ij} = -2\sum_{k=1}^{4} \log\left(p_{ij}^{k}\right)$, and p_{ij}^{k} denotes the p-value of the event that gene i is regulated by gene j from the kth dataset, and the final score for gene i to be regulated by gene j is given by $1 - P\left(X_8^2 \geq \chi_{ij}^2\right)$, where X_8^2 denotes a random variable that follows chi-square distribution with 8 degrees of freedom. Finally, we rank ordered the gene pairs based on the final score to infer the overall regulatory network.

22.4 An application to DREAM4 *in silico* Network Challenge

22.4.1 Background of DREAM4 in silico Data

The *in silico* network challenge of the fourth Dialogue for Reverse Engineering Assessments and Methods (DREAM4) provides biologically plausible simulated gene expression datasets in order to evaluate various reverse engineering methods in an unbiased manner. Further, because participants' performance is solely measured by prediction performance, more practical methods are encouraged to be developed. The detailed description of the challenge can be found at http://wiki.c2b2.columbia.edu/dream/ind ex.php/D4c2.

In the DREAM4 challenge, biologically plausible datasets were generated by first extracting network topologies from real transcriptional regulatory networks of *E. coli* and *S. cerevisiae*, where these extracted subnetworks may include cycles but not auto-regulatory loops. In the subsequent step, gene expression data were generated from these subnetworks using stochastic differential equations (SDEs), the Langevin equations [4], where dynamics for both mRNA and protein were specified by kinetics models. For proteins, the driving forces were given by the translation rate and degradation rate, whereas for mRNAs, the driving forces were described by the transcription rate and the degradation rate. The transcription rate was modulated by using thermodynamic models [6]. Hence, a complete regulatory model not only depends on the network structure, but also depends on the detailed conformation of the regulatory models [26, 43, 44].

To simulate the knock down and knock out effects on a gene, the transcription rate of this gene was reduced by half or down to zero, respectively. Independent knock down and knock out were performed for every gene in the network. Additionally, multi-factorial data were generated by slightly increasing or decreasing the basal activation of all genes in the network simultaneously with random magnitudes.

Multi-factorial perturbations were repeated as many times as the number of genes in the network. Each time course dataset has 21 time points, with four or five replicates. The initial condition always corresponds to wild type. A perturbation was applied to only one third of the genes during the first half of the time course, and was removed during the remaining time course.

The final datasets contain noise-added mRNA concentration levels (not protein concentration levels), and the noise is simulated from a mixture of normal and log normal distributions. All networks and data were generated using version 2.0 of GeneNetWeaver (GNW). Furthermore, datasets were normalized such that the maximum of normalized gene expression values for a given network is one.

The main goal of the challenge is to infer GRNs that are modeled by a directed graph, where each node represents a gene and each directed edge from gene A to gene B implies that gene A is a regulator of gene B. The challenge consists of one sub-challenge of 10-gene-network and two sub-challenges of 100-gene-network. For the 10-gene-network, all five previously described types of datasets (wild type, knock down, knock out, multi-factorial, and time course) were provided (sub-challenge 1). For the first 100-gene-network sub-challenge, all but multi-factorial data were provided (sub-challenge 2); and for the second sub-challenge, only the multi-factorial datasets were given (sub-challenge 3).

22.4.2 Reverse Engineering of DREAM4 in silico Data

We applied our method to DREAM4 *in silico* data, and the predicted networks were evaluated by the following two criteria: (1) AUPR: The area under the precision-recall curve, (2) AUROC: The area under the receiver-operator characteristics curve.

For the DREAM4 *in silico* challenge, we estimated wild type expression level for each gene in DREAM4 datasets by taking average of the first-time-point expression levels in the time course data plus the provided wild type, because the expression level at the first time point of each time course replicate can be considered as one wild type expression level according to the data description as well as our visual inspection of the time course data. We generated the sorted list of edges for the size 10 network challenge (sub-challenge 1) by using the method described in Sect. 22.3. For the one of the size 100 networks challenge (sub-challenge 2), the multi-factorial datasets were not given, and thus we were only able to combine three available datasets to generate the final score. For the other size 100 networks challenge (sub-challenge 3), where only multifactorial datasets were provided, we simply used RN method to generate the sorted list of edges.

We present the performance of our proposed method in Table 22.1 as well as that of three best performers in DREAM4 sub-challenges. We remark that the reported scores are provided by DREAM4 initiatives in an unbiased manner. Our approach was ranked second in the size 10 network challenge. Interestingly, the best performer of the sub-challenge 1 did not participate in the size 100 network challenges, and the best performers in the sub-challenges 2 and 3 did not show good performance

Table 22.1 Comparison of our proposed method to the best performers of DREAM4 *in silico* challenge

		AUPR/AUROC				
		Ecoli1	Ecoli2	Yeast1	Yeast2	Yeast3
Sub.1	Propsed method	0.881/0.967	0.382/0.796	0.682/0.916	0.698/0.902	0.424/0.822
	Best performer of Sub.1	0.916/0.972	0.547/0.841	0.968/0.990	0.852/0.954	0.761/0.928
	Best performer of Sub.2	0.590/0.764	0.225/0.606	0.681/0.830	0.767/0.928	0.406/0.703
	Best performer of Sub.3	0.629/0.852	0.285/0.680	0.458/0.852	0.595/0.808	0.400/0.710
Sub.2	Propsed method	0.427/0.906	0.379/0.777	0.314/0.835	0.309/0.848	0.105/0.766
	Best performer of Sub.1	n.a.	n.a.	n.a.	n.a.	n.a.
	Best performer of Sub.2	0.536/0.914	0.377/0.801	0.390/0.833	0.349/0.842	0.213/0.759
	Best performer of Sub.3	0.338/0.864	0.309/0.748	0.277/0.782	0.267/0.808	0.114/0.720
Sub.3	Propsed method	0.108/0.739	0.147/0.694	0.185/0.748	0.161/0.736	0.111/0.745
	Best performer of Sub.1	n.a.	n.a.	n.a.	n.a.	n.a.
	Best performer of Sub.2	0.130/0.698	0.110/0.636	0.194/0.722	0.170/0.724	0.162/0.708
	Best performer of Sub.3	0.154/0.745	0.155/0.733	0.231/0.775	0.208/0.791	0.197/0.798

Table 22.2 Comparison of our proposed method to the best performer of DREAM3 insilico challenge

	AUPR/AUROC				
	Ecoli1	Ecoli2	Yeast1	Yeast2	Yeast3
DREAM3 best performer	0.710/0.928	0.713/0.912	0.897/0.949	0.541/0.747	0.627/0.714
Propsed method	0.727/0.918	0.735/0.899	0.908/0.941	0.546/0.737	0.549/0.725

in the size 10 network challenge. Our method shows good performance across all of the sub-challenges, suggesting the robustness of our proposed method.

We also compare the performance of our method to that of the best performer in the DREAM3 challenge [50], using the 10-gene-network datasets from DREAM3. The DREAM3 *in silico* network challenge is the previous year's challenge of reverse engineering from heterozygous knock down, null mutants and trajectory datasets, which correspond to knock down, knock out and time course datasets, respectively, in DREAM4 challenge. Datasets were generated in a similar way to generate the datasets of DREAM4 except that sets of ODE were utilized for DREAM3. Because multi-factorial dataset was not provided in DREAM3 as in sub-challenge 2 of DREAM4, we were only able to combine three available datasets to generate the final score. In addition, in the DREAM3 time course data, perturbation is applied only at the first time point. We took the last data point of each time course replicate experiment and used the median of these values as the wild type estimate, assuming that the system is recovered to wild type at the end of the time course. The results are summarized in Table 22.2, and it appears that our method performs as well as the best performer in the DREAM3 *in silico* challenge.

To further investigate performance of our method, we simulated data by following the scheme outlined in Sect. 22.4.1. Since version 2 of GeneNetWeaver (GNW) was not available at the moment, we used version 1.2.0 of GNW, the simulation software used for DREAM3 challenge, to simulate topologies and regulatory parameters of 10-gene-networks. After true networks were generated, we integrated them into stochastic differential equations to get the dynamics of these networks. We simulated five types of data, i.e., wild-type, knock out, knock down, muti-factorial, and time series data, in line with the types of data provided by DREAM4 *in silico* network challenge. It is worth noting that for our simulation and DREAM4 challenge, SDE is used; whereas for DREAM3, ODE was utilized.

We analyzed a total of six network topologies (five from the provided true networks in DREAM3, and one from our own simulation setup). The new topology is provided in Fig. 22.3. For each topology, we generated five replicated datasets because the stochastic component of SDE might induce too much error in the dataset, which may disguise the true performance of our method. We also generated scores using merged data from different combinations of various experiments, to examine which combination would yield the best prediction.

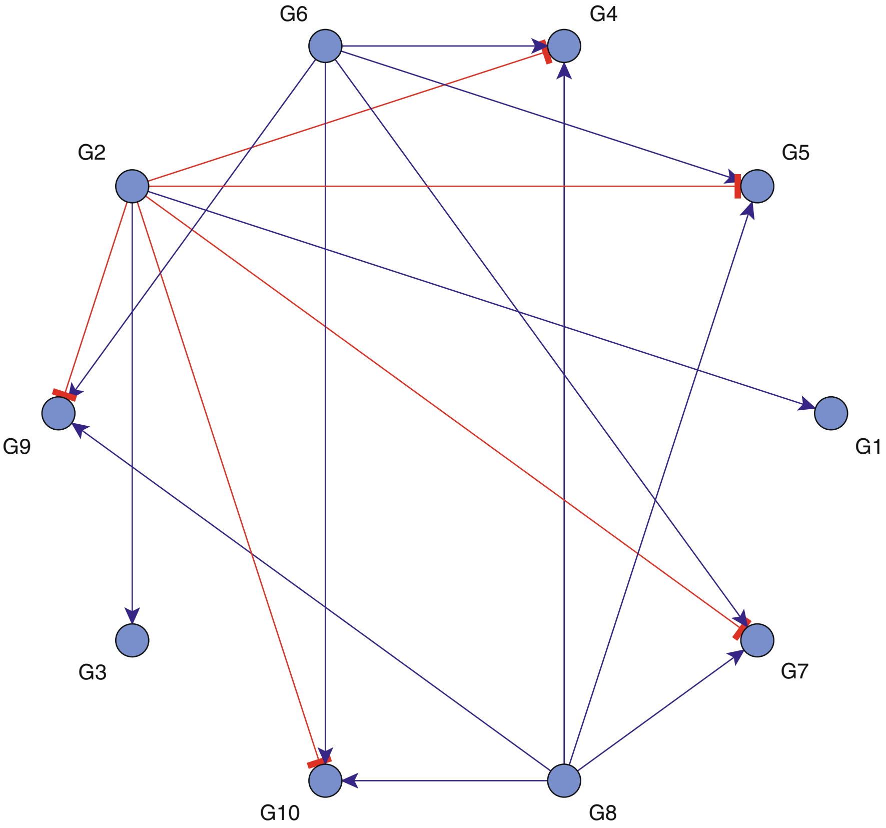


Fig. 22.3 Our newly simulated topology

Table 22.3 The effect of data aggregation of our method

(**a**) When internal noise level (standard error) of SDE is 0.01, and measurement error noise level (standard error) is 0.05

	AUPR/AUROC					
	Ecoli1	Ecoli2	Yeast1	Yeast2	Yeast3	New
All	0.662/0.905	0.745/0.921	0.855/0.954	0.502/0.675	0.530/0.676	0.649/0.830
Knockout only	0.631/0.819	0.796/0.937	0.861/0.959	0.469/0.645	0.509/0.640	0.685/0.825
Knockout + knockdown	0.606/0.841	0.785/0.926	0.838/0.941	0.492/0.659	0.527/0.664	0.641/0.827
Knockout+ mutifactorial	0.666/0.872	0.792/0.937	0.864/0.962	0.469/0.649	0.515/0.650	0.680/0.825
Knockout + time course	0.674/0.859	0.741/0.922	0.879/0.971	0.486/0.659	0.519/0.659	0.695/0.839

(**b**) When internal noise level (standard error) of SDE is 0.01, and measurement error noise level (standard error) is 0.1

	AUPR/AUROC					
	Ecoli1	Ecoli2	Yeast1	Yeast2	Yeast3	New
All	0.562/0.801	0.605/0.801	0.787/0.941	0.463/0.627	0.542/0.690	0.610/0.817
Knockout only	0.568/0.760	0.670/0.849	0.835/0.960	0.440/0.597	0.500/0.654	0.632/0.797
Knockout + knockdown	0.524/0.761	0.647/0.820	0.812/0.943	0.428/0.591	0.497/0.656	0.580/0.820
Knockout+ mutifactorial	0.551/0.774	0.652/0.839	0.783/0.934	0.441/0.598	0.501/0.653	0.628/0.784
Knockout + time course	0.614/0.803	0.625/0.825	0.804/0.953	0.476/0.639	0.563/0.716	0.670/0.828

(**c**) When internal noise level (standard error) of SDE is 0.05, and measurement error noise level (standard error) is 0.05

	AUPR/AUROC					
	Ecoli1	Ecoli2	Yeast1	Yeast2	Yeast3	New
All	0.345/0.693	0.413/0.665	0.365/0.752	0.393/0.646	0.332/0.584	0.419/0.724
Knockout only	0.320/0.638	0.468/0.734	0.385/0. 752	0.347/0.570	0.377/0.561	0.466/0.704
Knockout + knockdown	0.355/0.685	0.439/0.711	0.339/0.737	0.348/0.604	0.334/0.566	0.414/0.725
Knockout+ mutifactorial	0.284/0.637	0.435/0.697	0.351/0.737	0.364/0.602	0.370/0.566	0.432/0.705
Knockout + time course	0.356/0.655	0.439/0.699	0.407/0.790	0.405/0.641	0.401/0.612	0.472/0.707

In Table 22.3, we summarize the results from this study. Note that the values presented in the table are the average over the five replicates. We first observe that knock out data contain the most information on network topologies, suggesting an accurate estimate of wild type and noise model is important. Second, we find that utilizing multi-factorial and time-course data does improve AUPR and AUROC, implying that these datasets contain additional information to reduce false positives and false negatives. And finally, knock down data do not seem to benefit the prediction, presumably because they contain redundant but weaker information than that provided by the knock out data.

In summary, in the DREAM4 network challenge, our method performs second best in size 10 network challenge and also exhibits superior performance in the size 100 network challenges. In addition, using the datasets provided by DREAM3, we show that our proposed method performs as well as the best performer of the DREAM3 *in silico* challenge. Furthermore, using our simulated datasets, we demonstrate that network structures can be better predicted by combining multi-factorial, knock out, and time course data in the DREAM4 *in silico* challenge. But the additional benefit from combining the knock down dataset is not obvious.

22.5 Conclusion

In this article, we first reviewed the GRN inference methods, focusing on the methods that analyze gene expression data, including RNs, GGMs and BNs. Since there were various types of perturbation experiments to generate gene expression data, any single proposed GRN method cannot be optimal across various datasets. We thus proposed a new reverse engineering method that extracts experiment specific information and effectively integrates these multiple sources of information. One unique feature of our approach is to combine various datasets across experiments and facilitate the statistical modeling of background noise by assuming that the noise from each gene's mRNA measurements across experiments follows the same normal distribution with mean 0 and variance σ_i^2, for each gene i.

From simulation studies, we observe that most of the information on network structure can be inferred from knock out data alone; and the limitation of using only knock out data to perform parameter estimation can be reduced by incorporating additional information from multi-factorial and time course datasets. However, the knock down data do not seem to benefit the inference of network structure in our method. This suggests two future directions. The first is to devise a method to more efficiently combine the evidences so that the contribution from weak signals can be weighed down, whereas that from strong signals can be weighed up; and the second is to design a more sophisticated algorithm to better extract the information from the knock down data.

The structure of a directed graph is assumed for the network topology, however, our use of multi-factorial and time-course data does not carefully consider the characteristics of a directed graph. First of all, the correlation measure, used to generate the score for the multi-factorial dataset, does not have any directional information. Thus, it gives equal scores to edges in both directions. Although this may not represent the best way of inferring directed edges, it should not seriously harm the accuracy of inference either. However, a more serious problem may arise in our current usage of time series data, where a naive conditional approach is implemented to differentiate direct from indirect edges without fully exploiting the structure of the graph (e.g. V-structure [24]), which could lead to aberrant predicted network structures. Although the effects of the aforementioned theoretical shortcomings of our method do not seem to be profound in our simulation; in the future, we would like to design a method to more accurately differentiate direct from indirect edges to better address these issues.

Acknowledgements Supported in part by NIH grants GM59507 and T15LM07056, a pilot grant from the Yale Pepper Center, NSF grant DMS0714817, and a fellowship from the China Scholarship Council.

References

1. Andercut, M., & Kauffman, S. A. (2008). On the sparse reconstruction of gene networks. *Journal of Computational Biology*, *15*(1), 21–30.
2. Andrecut, M., Huang, S., & Kauffman, S. A. (2008). Heuristic approach to sparse approximation of gene regulatory networks. *Journal of Computational Biology*, *15*(9), 1173–1186.
3. Bansal, M., Gatta, G. D., & di Bernardo, D. (2006). Inference of gene regulatory networks and compound mode of action from time course gene expression profiles. *Bioinformatics*, *22*(7), 815–822.
4. Bernt Øksendal, (2006). *Stochastic differential equation: An introduction with applications* (6th ed.). Springer Heidelberg Dordrecht London New York.
5. Bonneau, R., Reiss, D. J., Shannon, P., Facciotti, M., Hood, L., Baliga, N. S., & Thorsson, V. (2006). The inferelator: An algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biology*, *7*, R36.
6. Buchler, N. E., Garland, U., & Hwa, T. (2003). On schemes of combinatorial transcription logic. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(9), 5136–5141.
7. Butte, A. J., Tamayo, P., Slonim, D., Golub, T. R., & Kohane, I. S. (2000). Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. *Proceedings of the National Academy of Sciences of the United States of America*, *97*(22), 12182–12186.
8. Butte, A. S., & Kohane, I. S. (2003). Relevance networks: A first step toward finding genetic regulatory networks within microarray data. In G. Parmigiani, E.S. Garrett, R.A. Irizarry, & S.L. Zeger (Eds.), *The analysis of gene expression data: Methods and Software*, (pp. 428–446). New York, NY: Springer.
9. Cao, J., & Zhao, H. (2008). Estimating dynamic models for gene regulation networks. *Bioinformatics*, *15*;24(14), 1619–1624.
10. Chen, K. C., Wang, T. Y., Tseng, H. H., Huang, C. Y. F., & Kao, C. Y. (2005). A stochastic differential equation model for quantifying transcriptional regulatory network in saccharomyces cerevisiae. *Bioinformatics*, *21*(12), 2883–2890.
11. Chickering, D. M. (1996). Learing Bayesian Networks is NP-complete. In D. Fisher & H.-J. Lenz (Eds.), *Learning from data: Artificial intelligence and statistics*, (pp. 121–130). New York: Springer-Verlag.
12. Christley, S., Nie, Q., & Xie, X. (2009). Incorporating existing network information into gene network inference. *PLoS ONE*, *4*(8), e6799.
13. Chu, S., DeRisi, J., Eisen, M., Botstein, J., Brown, P. O., & Herskowitz, I. (1998). The transcriptional program of sporulation in budding yeast. *Science*, *282*(5389), 699–705.
14. Dobra, A., Hans, C., Jones, B., Nevins, J. R., Yao, G., & West, M. (2004). Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, *90*(1), 196–212.
15. Ellis, B., & Wong, W. H. (2008). Learning causal Bayesian network structures from experimental data. *Journal of the American Statistical Association*, *103*, 778–789.

16. Fisher, R. A. (1948). Combining independent tests of significance. *American Statistician*, *2*(5), 30.
17. Fisk, P. R. (1970). A note on a characterization of the multivariate normal distribution. *The Annals of Mathematical Statistics*, *41*, 486–494.
18. Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, *7*, 601–620.
19. Gardner, T. S., di Bernardo, D., Lorenz, D., & Collins, J. J. (2003). Inferring genetic networks and identifying compound mode of action via expression profiling. *Science*, *4*, 102–105.
20. Gasch, A. P., Spellman, P. T., Kao, C. M., Carmel-Harel, O., Eisen, M. B., Storz, G., Botstein, D., & Brown, P. O. (2000). Genomic expression programs in the response of yeast cells to environmental changes. *Molecular Biology of the Cell*, *11*(12), 4241–4257.
21. Granger, C. W. J. (1980). Testing for causality, a personal viewpoint. *Journal of Economic Dynamics and Control*, *2*, 329–352.
22. Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., & Young, R. A. (2001). Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *Pac Symp Biocomput.*, 422–433.
23. Hughes, T. R., Marton, M. J., Jones, A. R., Roberts, C. J., Stoughton, R., Armour, C. D., Bennett, H. A., Coffey, E., Dai, H., He, Y. D., Kidd, M. J., King, A. M., Meyer, M. R., Slade, D., Lum, P. Y., Stepaniants, S. B., Shoemaker, D. D., Gachotte, D., Chakraburtty, K., Simon, J., Bard, M., & Friend, S. H. (2000). Functional discovery via a compendium of expression profiles. *Cell*, *102*(1), 109–126.
24. Jordan, M. I. (2004). Graphical models. *Statistical Science*, *19*(1), 140–155.
25. Li, H., & Gui, J. (2006). Gradient directed regularization for sparse gaussian concentration graphs, with applications to inference of genetic networks. *Biostatistics*, *7*(2), 302–317.
26. Marbach, D., Schaffter, T., Mattiussi, C., & Floreano, D. (2009). Generating realistic in silico gene networks fro performance assessment of reverse engineering methods. *Journal of Computational Biology*, *16*(2), 229–239.
27. Margolin, A. A., Memenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R. D., & Califano, A. (2006). ARCANE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, *7*(1), S7.
28. Meinshausen, N., & Buhlmann, P. (2006). High-dimensional graphs and variable selection with the LASSO. *Annals of Statistics*, *34*(3), 1436.
29. Opgen-Rhein, R., & Strimmer, K. (2007). Learning causal networks from systems biology time course data: An effective model selection procedure for the vector autoregressive process. *BMC Bioinformatics*, *8*(2), S3.
30. Peng, J., Wang, P., Zhou, N., & Zhu, J. (2009). Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, *104*(486), 735–746.
31. Perrin, B. E., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J., & d'Alche-Buc, F. (2003). Gene networks inference using dynamic Bayesian networks. *Bioinformatics*, *19*(2), ii138–ii148.
32. Ramsay, J. O., Hooker, G., Cao, G., & Campbell, D. (2007). Parameter estimation for differential equations: A generalized smoothing approach (with discussion). *Journal of Royal Statistical Society, Series B*, *69*, 741–769.
33. Rice, J. J., Tu, Y., & Stolovizky, G. (2005). Reconstructing biological networks using conditional correlation analysis. *Bioinformatics*, *21*(6), 765–773.
34. Rogers, S., & Girolami, M. (2005). A Bayesian regression approach to the inference of regulatory networks from gene expression data. *Bioinformatics*, *21*(14), 3131–3137.
35. Savageau, M. A. (1969). Biochemical systems analysis. I. some mathematical properties of the rate law for the component enzymatic reactions. *Journal of Theoretical Biology*, *25*(3), 365–369.
36. Savageau, M. A. (1969). Biochemical systems analysis. II. the steady-state solutions for an n-pool system using a power-law approximation. *Journal of Theoretical Biology*, *25*(3), 370–379.
37. Schafer, J., & Strimmer, K. (2005). An empirical Bayes approach to inferring large-scale gene association networks. *Bioinformatics*, *21*, 754–764.

38. Schafer, J., & Strimmer, K. (2005). A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. *Statistical Applications in Genetics and Molecular Biology*, *4*, 1–30.
39. Shimamura, T., Imoto, S., Yamaguchi, R., Fujita, A., Nagasaki, M., & Miyano, S. (2009). Recursive regularization for inferring gene networks from time-course gene expression profiles. *BMC Systems Biology*, *3*, 41.
40. Shimamura, T., Imoto, S., Yamaguchi, R., & Miyano, S. (2007). Weighted LASSO in graphical Gaussian modeling for large gene network estimation based on microarray data. *Genome Informatics*, *19*, 142–153.
41. Soranzo, N., Bianconi, G., & Altafini, C. (2007). Comparing association network algorithms for reverse engineering of large-scale gene regulatory networks: Synthetic versus real data. *Bioinformatics*, *23*, 1640–1647.
42. Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K., Eisen, M. B., Brown, P. O., Botstein, D., & Futcher, B. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Molecular Biology of the Cell*, *9*(12), 3273–3297.
43. Stolovitzky, G., Monroe, D., & Califano, A. (2007). Dialogue on reverse-engineering assessment and methods: The DREAM of high-throughput pathway inference. *Annals of the Newyork Academy of Sciences*, *1115*, 1–22.
44. Stolovitzky, G., Prill, R., & Califano, A. (2009). Lessons from the DREAM2 challenges. *Annals of the New York Academy of Sciences*, *1158*, 159–195.
45. Sun, N., & Zhao, H. (2009). Reconstructing transcriptional regulatory networks through genomic data. *Statistical Methods in Medical Research*, *18*, 595–617.
46. Wahba, G. (1985). A comparison of GCV and GML for choosing the smoothing parameter in the generalized spline smoothing problem. *Annals of Statistics*, *13*, 1378–1402.
47. Wang, S. C. (2004). Reconstructing gene networks from tiem ordered gene expression data using bayesian method with global search algorithm. *Journal of Bioinformatics and Computational Biology*, *2*, 441–458.
48. Werhli, A. V., Grzegorczyk, M., & Husmeier, D. (2006). Comparative evaluation of reverse engineering gene regulatory networks with relevance networks, graphical gaussian models and Bayesian networks. *Bioinformatics*, *22*, 2623–2531.
49. Wille, A., & Buhlmann, P. (2006). Low-order conditional independence graphs for inferring genetic networks. *Statistical Applications in Genetics and Molecular Biology*, *5*(1), Article 1.
50. Yip, Y. L. (2009). Computational reconstruction of biological networks. Ph.D. thesis, Yale University, New Haven, CT.
51. Zou, M., & Conzen, S. D. (2005). A new dynamic bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics*, *21*, 71–79.

Chapter 23
Inferring Signaling and Gene Regulatory Network from Genetic and Genomic Information

Zhidong Tu, Jun Zhu, and Fengzhu Sun

Abstract Biological systems respond to environmental changes and genetic variations. One of the essential tasks of systems biology is to untangle the signaling and gene regulatory networks that respond to environmental changes or genetic variations. However, unwiring the complex gene regulatory program is extremely challenging due to the large number of variables involved in these regulatory programs. The traditional single gene centered strategy turns out to be both insufficient and inefficient for studying signaling and gene regulatory networks. With the emergence of various high throughput technologies, such as DNA microarray, ChIP-chip, etc., it becomes possible to interrogate the biological systems at genome scale efficiently and cost effectively. As these high throughput data are accumulating rapidly, there exists a clear demand for methods that effectively integrate these data to elucidate the complex behaviors of biological systems. In this chapter, we discuss several recently developed computational models that integrate diverse types of high throughput data, particularly, the genetic and genomic data, as examples for the systems approaches that untangle signaling and gene regulatory networks.

23.1 Signaling and Regulatory Networks in Biological Systems

Living organisms need to constantly adjust themselves to cope with environmental changes and changes happened within themselves. For example, yeast cells need to quickly adapt to different nutrition sources in order to survive in the wild

Z. Tu
Merck & Co., Inc., Boston, MA, USA
e-mail: zhidong_tu@merck.com

J. Zhu
Sage Bionetwork, Seattle, WA, USA
e-mail: jun.zhu@sagebase.org

F. Sun (✉)
University of Southern California, Los Angeles, CA, USA
e-mail: fsun@usc.edu

Zhidong Tu and Jun Zhu contributed equally to this chapter

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_23,
© Springer-Verlag Berlin Heidelberg 2011

environment, and human cells in various tissues need to divide and grow at different speed and timings to ensure normal body development. All these are accomplished by complex signaling and regulatory networks which allow the system to sense the change and adjust its behavior accordingly. Many human diseases are associated with the defects in these signaling and regulatory networks. For example, global survey of phosphotyrosine signaling identified abnormal oncogenic kinase activation in lung cancer [1], and multiple oncogenic pathway signatures showed coordinated expression changes in prostate tumors [2]. It is clear that decoding these signaling and regulatory networks is critical for understanding the biological systems and important for developing new treatments for human diseases caused by the disruptions of these networks.

We use glucose signaling pathway in yeast as an example for signaling and regulatory networks. Yeast cells prefer fermentable carbon source to nonfermentable carbon source that has to be metabolized by oxidation [3]. Addition of glucose to yeast cells growing on a nonfermentable carbon source triggers rapid global changes to the system to allow quick utilization of the favorable energy source. More than 40% of the genes' expression changes more than two folds within minutes following addition of glucose [4]. As shown in Fig. 23.1, multiple pathways are involved in glucose signaling. They coordinate with each other to sense and integrate the glucose level signal, and respond by transcriptional regulation of a large pool of genes [5] and activating or repressing certain enzymes (not shown in the figure). As highlighted in the figure, majority of the transcription regulations occur via pathways intermediated by GTP binding proteins Ras and Gpr/Gpa2, whose activation lead to rapid increase of intracellular cAMP, this in turn activates protein kinase A (PKA), which is responsible for re-programming of thousands of genes' expression by phosphorylating its downstream targets including multiple transcription factors (TFs) [6].

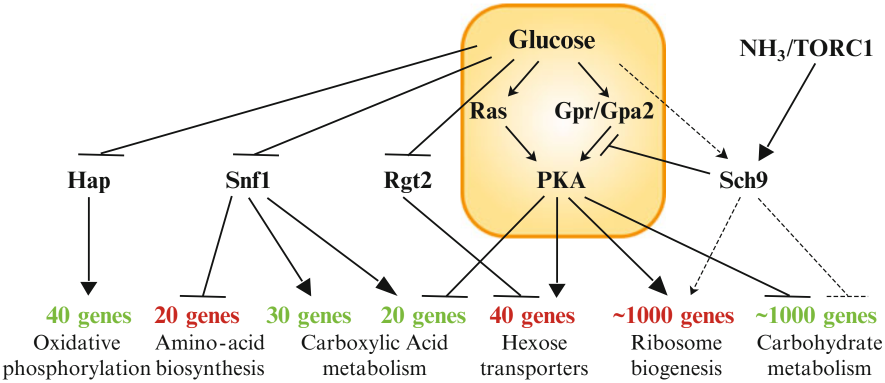


Fig. 23.1 Diagram of the regulatory wiring connecting the addition of glucose to the transcriptional responses of the cell. *Dotted lines* indicate a limited or indirect connection (reproduced and modified from [5])

23.2 Genetic Variation and Gene Expression Regulation

Perturbations on certain pathway nodes (e.g., PKA in Fig. 23.1) have been experimentally shown to cause changes to downstream genes' expression [5]. Here, we study one category of perturbations, i.e., the naturally occurring DNA variations, and their impact on gene expression. By treating a gene expression as a classic phenotypic trait, linkage analysis has been applied to identify the genetic factors that lead to gene expression variations. From simple organisms (e.g., budding yeast), to most advanced mammals such as human and mouse, researchers have consistently identified numerous genetic factors that regulate the expression of a large number of genes [7–9]. As some of these genetic factors are linked to disease phenotypes by linkage analysis, and genes regulated by these genetic factors have been shown to lead to disease development when knocked out in mouse models [10, 11], studying the networks underlying these genetic perturbations provides a novel approach for understanding the diseases at unprecedented systems level.

Again, we use yeast as the model organism to describe systems approaches of dissecting the genetic landscape of global gene expression regulation, we then discuss the most recent development on higher organisms at the end of this chapter.

Brem and Kruglyak performed a pioneering study on genetic mapping of global gene expression measured by microarrays in 2002 [8]. In that experiment and the following ones [12, 13], they crossed two strains of budding yeast *Saccharomyces cerevisiae*, a standard laboratory strain (BY) and a wild strain isolated from a California vineyard (RM) (Fig. 23.2a). Over one hundred segregants from the cross were then profiled for their genotypes and global gene expression levels.

The two parental yeast stains have quite different global gene expression profiles. A total of 1,528 genes show differential expression at $P < 0.005$, whereas only 23 are expected by chance [8]. More interestingly, expression measurements in haploid segregants suggest that parental differences in expression are highly heritable.

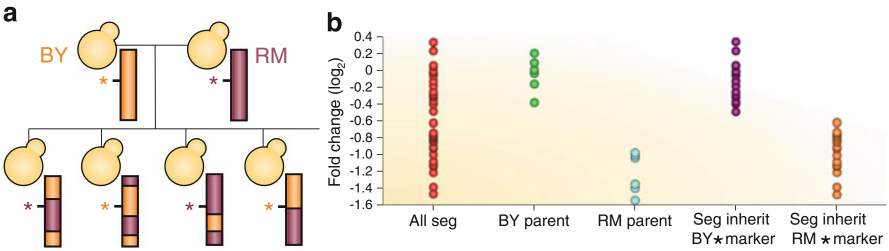


Fig. 23.2 Two yeast strains (BY and RM) were crossed and multiple segregants were cultured and profiled for both genotype and global gene expression. (**a**) the *rectangle* indicates yeast chromosome and * indicates genetic marker. Different *colors* represent different allele types. Segregants inherited different allele types from either BY or RM at the marker position, (**b**) an example showing the linkage between gene expression trait and the genetic marker. (Reproduced and modified from [14])

As illustrated in Fig. 23.2b, the distribution of a particular gene's expression is tightly related to the allele type measured at genetic marker * and is similar to the distribution in the parental strain carrying the same allele.

Clearly when treated as classic traits, certain gene expression levels can be linked to chromosomal regions based on linkage analysis. These mapped regions on chromosomes are called expression trait loci or eQTLs. Determining eQTLs is obviously an important step towards dissecting the genetic structures that regulate gene expression. However, these eQTLs often contain large number of genes and makes it difficult to unambiguously identify the factors that cause expression variation. Secondly, even if the eQTL contains very few genes and the causal genetic factors can be identified by including additional clues and/or reasoning, the mechanisms for the factors to cause expression variation remain unclear. In the next section, we introduce several approaches addressing these problems.

23.3 Inferring Causal Genes and Regulatory Networks

To solve the problems mentioned above, several integrative computational methods have been developed and demonstrated to be effective in identifying the causal regulators and elucidating the underlying regulatory networks. We examine two approaches in details in the following sections, and briefly discuss several related works at the end.

23.3.1 Approach I: Identifying Signaling and Regulatory Paths

Tu et al. proposed an approach that aims at identifying signaling and regulatory paths (or gene networks) in the biological system that link the genetic factor in the eQTL with the expressionally perturbed genes (called target genes). Since the perturbation of nodes on such regulatory paths is linked to target gene expression variations, identifying these paths would help to reveal the underlying regulatory mechanisms and facilitate the causal gene inference.

As shown in Fig. 23.3, the approach works by first constructing a gene network, which consists of protein-protein interactions, protein phosphorylation and TF-DNA biding information. Presumably, this gene network contains general knowledge of interactions within the system that are involved in signaling and expression regulation. However, having this global gene network does not directly answer which genes in eQTLs are responsible for target gene expression variation. To provide a solution, the algorithm tries to identify paths in the gene network that connects candidate genes physically resided in eQTLs and genes that are linked to the eQTLs given certain assumptions.

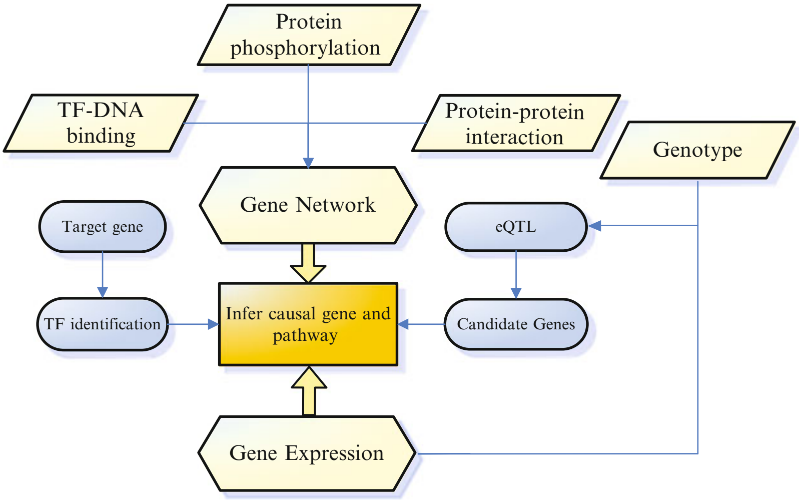


Fig. 23.3 Overview of the procedures for causal gene identification and regulatory pathway inference by Tu et al. (figure reproduced from [15])

23.3.1.1 Basic Assumptions

Two basic assumptions are made by Tu et al. [15]. First, we assume that the causal gene regulates the target genes by modulating the activities of TFs of the corresponding target genes. Although multiple mechanisms are responsible for regulating gene expression (e.g., by regulating mRNA degradation, transcription rate, etc.), regulation on transcription factor's activity is commonly regarded as the dominant form. Second, we assume that the activities of genes on the pathway correlate with target gene's expression. The concept is illustrated in Fig. 23.4, where the target gene's expression is affected by the activity of node on the pathway. As current high throughput technology can not directly measure protein activity, we use gene's expression as an approximation. Although this approximation may be unreliable at individual gene's level, it is acceptable when considering the general trend at genome scale [16] and is a common practice for many microarray data analyses. Pathway-wise gene-gene correlation was studied by Zien et al. and their results suggested that genes on the same pathway were more synchronized in their expression levels [17]. The second assumption is important as it indicates that not all the paths in the gene network are equally relevant to specific target genes, and ranking these paths can be done based on expression correlation of genes in the path.

23.3.1.2 Identifying Regulatory Paths in the Network

At the kernel of Tu et al.'s approach, efforts are spent on finding paths in the network that connect genes physically residing within eQTLs and target genes whose

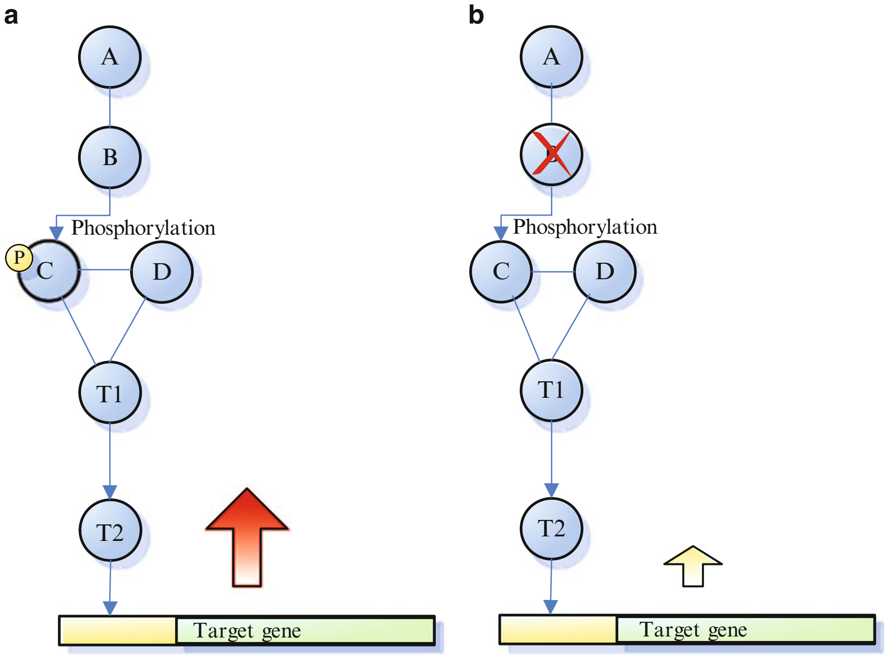


Fig. 23.4 A conceptual gene regulatory pathway in activated status (**a**) and deactivated status (**b**). Nodes in *large circles* are proteins; in particular, T1 and T2 are transcription factors. In (**a**), B is actively phosphorylating protein C, and as this signal passes down to the pathway, target gene is actively expressed. (**b**) due to mutations in B, the pathway is no longer in active status and target gene's expression is at residual level. It is noteworthy that the pathway does not need to be strictly simple linear, as D could interact with C and D to form complex regulatory network. (figure reproduced from [15])

expression linked to the same eQTLs so that expression of the genes on these paths are more correlated with the target gene expression than what would be expected from randomly selected paths. However, as causal gene is unknown and needs to be identified, all genes physically residing in eQTLs are considered as candidates and ranked by certain scoring functions.

Given a target gene and its eQTL, Tu et al. initiate "walks" in the network starting from TFs that bind to the promoter region of the target gene. Once arrived at a node, the next node to visit is chosen stochastically by favoring genes whose expressions are highly correlated with target gene g_t (Fig. 23.5). Some walks will eventually arrive at genes within eQTL and these genes will be visited at different frequencies when walks are repeated multiple times. The algorithm can be formalized as follows.

For a target gene g_t, the set of transcription factors binding to its promoter region are denoted as $T_{g_t} = (t_1, \ldots, t_n)$, and the candidate causal genes in the eQTL regions are denoted as $C_{g_t} = (g_{c_1}, \ldots, g_{c_m})$. The gene network is represented as a graph G in which the protein-protein interactions are represented as

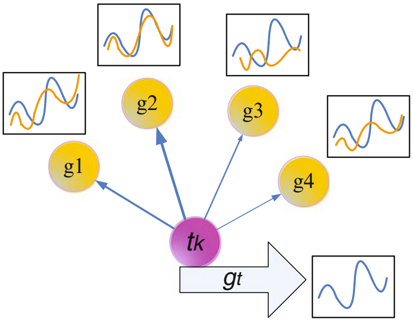


Fig. 23.5 Initial step for "walking" in the network. TFs for the target gene are identified based on TF-DNA binding data and are taken as initiating points for the walk. The next gene to visit is selected stochastically from all neighbors based on expression correlation with the target gene. Genes with stronger correlation are favored for being chosen as next node to visit (see main text for details)

undirected edges while protein phosphorylation and TF-DNA bindings are represented as directed edges. For each $t_k \in T_{g_t}$, we start a stochastic search procedure as shown in Fig. 23.6.

We denote all the neighbors of a particular gene in the gene network as $Nei(\cdot)$, so that $b \in Nei(a) \Leftrightarrow e_{ba} \in G$, where e_{ba} represents a directed edge from b to a. Starting from t_k, we estimate for each $g_i \in Nei(t_k)$ the "likelihood" that g_i is causative for the expression variation of the target gene g_t. Based on our second assumption, we estimate such causal effect by the absolute value of the Pearson correlation coefficient of g_i and g_t expression levels, denoted as $|\rho_{(g_i,g_t)}|$. Intuitively, a gene with strong expression correlation with the target gene is more likely to be involved in the same pathway. However, as not all genes on the pathway necessarily correlate with the target gene due to other post-translational regulation mechanisms, we give non-correlated genes a residual probability for being on the regulatory pathway by defining the casual effect of g_i with respect to g_t as $\xi(g_i, g_t) = \max\{|\rho(g_i, g_t)|, \varepsilon\}$, where $0 < \varepsilon < 1$ is the residual causal effect that a non-correlated gene could have upon g_t.

We denote a path as $P(g_0, g_1, \ldots, g_z)$, where $g_0, g_1, \ldots, g_z$ are nodes in the graph and cycles are disallowed in the path, i.e., $g_i \neq g_j$ for any g_i, g_j on the path. To ensure paths are non-cyclic, a set U is introduced which contains only unvisited genes. We stochastically select $g_i \in Nei(t_k) \cap U$ and transit from t_k to g_i. The transition probability is determined by Eq. 23.1.

Unvisited neighbor genes will be randomly drawn according to this transition probability. The chosen gene will be removed from U thereafter.

$$\Pr\{g_i | t_k, g_i \in Nei(t_k) \cap U\} = \frac{\xi(g_i, g_t)}{\sum\limits_{g_s \in Nei(t_k) \cap U} \xi(g_s, g_t)} \tag{23.1}$$

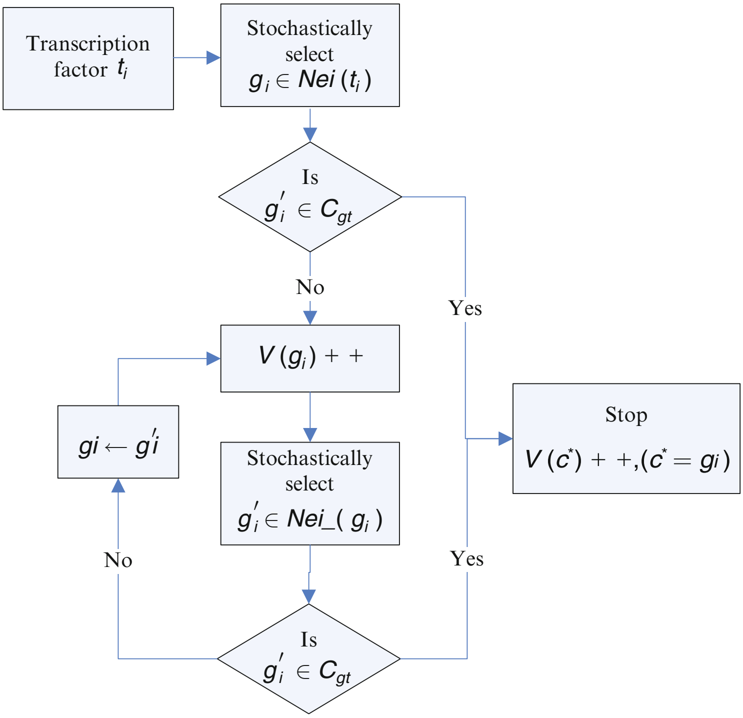

Transcription factor t_i
Stochastically select $g_i \in Nei(t_i)$
Is $g_i' \in C_{gt}$
No
Yes
$V(g_i)++$
$gi \leftarrow g'i$
Stochastically select $g_i' \in Nei\_(g_i)$
Stop
$V(c^*)++,(c^* = gi)$
No
Yes
Is $g_i' \in C_{gt}$

Fig. 23.6 The flow diagram of the stochastic searching algorithm

After we arrive at g_i, the same procedure is repeated. We select $g_i' \in Nei(g_i) \cap U$ based on similar transition probability as described by Eq. 23.2.

$$\Pr\{g_i' | g_i, g_i' \in Nei(g_i) \cap U\} = \frac{\xi(g_i', g_t)}{\sum_{g_s \in Nei(g_i) \cap U} \xi(g_s, g_t)} \tag{23.2}$$

It is of note that the algorithm always calculates the causal effect of a gene g_i with respect to g_t, which is different from most transcription regulatory network inference algorithm. In this procedure, the objective is not to identify the relationship between connected genes (i.e., g_i and g_i'), but to find connected genes which are likely to be causative for the expression variation of the target gene g_t.

The above procedure stops when it reaches any gene $g_i \in C_{g_t}$ or when it enters a dead end (i.e., $Nei(g_i) \cap U = \emptyset$). We also set an upper bound for the total number of transitions allowed to ensure a stop. The upper bound is chosen to be unrealistically high for any known pathway and is different from the path length in those deterministic pathway finding algorithms. Suppose we stop at $g_c \in C_{g_t}$ after one round of the procedure, the path can be written as $P(t_k, \ldots, g_i, \ldots, g_c)$. The causal effect of g_c on g_t through $P(t_k, \ldots, g_i, \ldots, g_c)$ can be calculated by Eq. 23.3. Here, we assume that the causal effect of each node on the pathway is independent with each other. This assumption may not always hold. However, considering interactions among genes on the pathway will make the problem too complex and we do not consider them in this chapter.

$$p(g_c, t_k, P(t_k, \ldots, g_c)) = \xi(t_k, g_t) \cdot \cdots \cdot \xi(g_c, g_t). \tag{23.3}$$

Equation 23.3 measures the causal effect of g_c on g_t with respect to a specific potential pathway, and the general causal effect of g_c considering the whole gene network can be estimated by Eq. 23.4, where $P_{t_k}^{g_c}$ denotes all the paths starting from t_k and ending at g_c.

$$p(g_c, t_k) = \sum_{P_{t_k}^{g_c}} p(g_c, t_k, P(t_k, \ldots, g_i, \ldots, g_c)). \tag{23.4}$$

To calculate $p(g_c, t_k)$, each gene $g_i \in G$ is associated with a counter $V_{t_k}(g_i)$ to record the times it has been visited. We iterate the whole procedure N times and N is set to be large enough so that (23.5) can be approximated, where $V_{t_k}(g_c)$ denotes the visit times for $g_c \in C_{g_t}$.

$$\lim_{N \to +\infty} V_{t_k}(g_c)/N = p(g_c, t_k). \tag{23.5}$$

If the target gene has more than one TF, each TF is assigned a weight based on its causal effect on the target gene and is linearly combined as shown by Eq. 23.6. The probability that g_c is the casual gene in the eQTL considering all the TFs for the target gene g_t is estimated by Eq. 23.7.

$$V_T(g_c) = \frac{\sum\limits_{k=1}^{m} \xi(t_k, g_t) V_{t_k}(g_c)}{\sum\limits_{k=1}^{m} \xi(t_k, g_t)} \tag{23.6}$$

$$\widehat{\Pr(g_c)} = \frac{V_T(g_c)}{\sum\limits_{g_s \in C(g_t)} V_T(g_s)} = \frac{\sum\limits_{k} p(g_c, t_k^c)}{\sum\limits_{s: g_s \in C(g_t)} \sum\limits_{k} p(g_s, t_k^s)} \tag{23.7}$$

Assuming there is only one causal gene in each eQTL, the gene with the largest posterior probability is reported as the causal gene as shown by (23.8).

$$g_c^* = \arg\max_{g_s \in C_{g_t}} \widehat{\Pr(g_s)} \tag{23.8}$$

To identify the underlying pathway, we start from g_c^* and trace backwards. We find from $Nei(g_c^*)$ the gene with the largest visit count and move to that gene (not stochastically). We repeat until we arrive at t_k. By this way, we find the most probable pathway which links g_c^* and t_k. The linear pathway generated by this approach is mainly for simplicity consideration. As indicated by Eq. 23.4, there could be multiple paths connecting g_c^* and t_k, and all of them contribute to the causal effect of g_c^*.

23.3.1.3 Testing with Yeast Knockout Compendium Data

To objectively measure the performance of the approach, Tu et al. designed a testing scheme using Rosetta yeast knockout compendium data [18]. From knockout expression profiles, Tu et al. used their algorithm to infer genes that were knocked out. Since the knockout genes are known, the prediction accuracy is obtainable. To transform this test into the same problem as eQTL causal gene inference, several major steps were proposed and are listed below:

1. Identify genes whose expression is significantly perturbed for each deletion mutation experiment and treat these genes as target genes of the knockout gene.
2. For each knockout, simulate an eQTL region around the deleted gene so the region contains ten genes. These ten genes position consecutively on the same chromosome and the deleted gene is randomly positioned at position one to ten.
3. Run the algorithm to identify the knockout gene from the ten genes.
4. Calculate the overall prediction accuracy. The method is expected to have higher than 10% correct prediction rate if it performs better than random guess.

Tu et al. reported an overall accuracy rate of 46%, which is four times better than random guess. This suggests that such integrative approach does help to identify the underlying genetic factors that are responsible for the expression changes in the system. However, due to incomplete information for TF-DNA binding, protein-protein interaction, etc., the coverage of such predictions is not particularly high with only ∼30% knockouts being considered as predictable by the algorithm.

23.3.1.4 An Example of Predicted Causal Genes and Inferred Regulatory Network

We show one example of such inferred regulatory network generated from the algorithm. For gene PRP39, a component of RNA splicing factor U1 small nuclear ribonucleoprotein polypeptide, its eQTL on chromosome VIII consists of three genes (see Fig. 23.7). From chromatin immunoprecipitation (ChIP) experiments, two TFs (DIG1 and STE12) bind to the promoter region of PRP39. The algorithm reports the same gene (GPA1) as the causal gene when it initiated with either of the two TFs. There are other genes linked to the same eQTL (e.g., FAR1) with the same inferred causal gene and pathway. Many of these genes are known to be involved in pheromone signaling [19]. By comparing the pathway identified by the algorithm (Fig. 23.7) with known pheromone pathway, large fraction of proteins are matched and arranged in the correct order. Furthermore, Yvert et al. performed an experiment

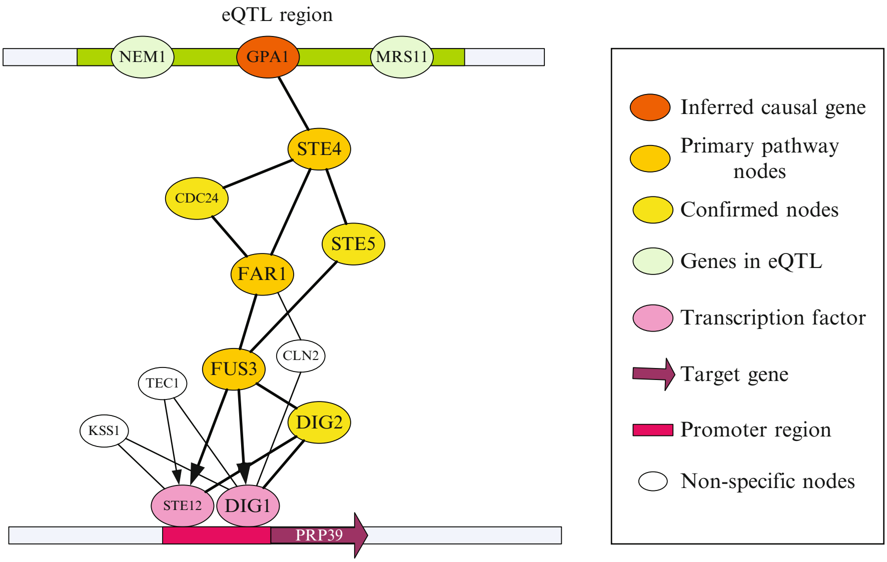


Fig. 23.7 An example of inferred causal gene and its associated regulatory network. Edges without *arrows* are protein-protein interactions. Edges with *arrow* represent phosphorylation or TF-DNA binding. Only nodes been visited more frequently than GPA1 and having at least two interactions with primary pathway nodes are shown

by making a point mutation in GPA1 in one of the yeast strains and observed that those downstream genes displayed altered expression levels as expected [20], which confirms GPA1's role in causing expression variation of downstream genes.

23.3.2 Approach II: Constructing De Novo Causal Networks

In this section, we discuss an alternative approach that constructs networks de novo based on gene expression, protein-protein interaction, TF-DNA binding information, metabolite profiles, literature information and other sources of information [21]. The overview of this method is illustrated in Fig. 23.8.

23.3.2.1 Causal Inferences

The first step of constructing causal networks is to infer pair-wise causal/reactive relationship under perturbations. Biological systems change dynamically in response to genetic or environment perturbations. Multiple methods have been proposed to model the causal/reactive relationships underlying the dynamic behaviors of the systems, such as Granger causality test based on time series [22], Dynamic Bayesian networks [23] and differential equations [18]. As there are feedback

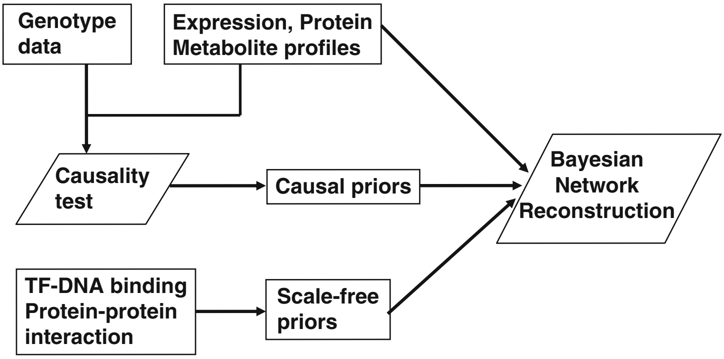

Genotype data
Expression, Protein Metabolite profiles
Causality test
Causal priors
Bayesian Network Reconstruction
TF-DNA binding Protein-protein interaction
Scale-free priors

Fig. 23.8 The flow diagram of the Bayesian network reconstruction process. High throughput data, such as gene expression profiles, protein profiles and metabolite profiles are used as main data for Bayesian network reconstruction. Genotype data, transcription factor binding site data and protein-protein interactions are used to generate different structure priors

regulations in biological systems, the systems will reach semi-static states eventually after constant perturbations. One example of such constant perturbations is genetic perturbation depicted in Fig. 23.2 above. Using static state data alone, e.g., gene expression data, we can not distinguish causal relationships, $A \rightarrow B \rightarrow C$, $A \leftarrow B \rightarrow C$ and $A \leftarrow B \leftarrow C$, which are Markov equivalent. If A is genotype at a locus L and gene expression trait or metabolic traits can not affect genotypes in general, then only one structure in the Markov equivalent class, $L \rightarrow B \rightarrow C$, is possible so that we can make causal inference about the order of B and C with regard to L [11]. There are two steps in the causality test procedure: pleiotropy test and causal model selection.

23.3.2.2 Pleiotropy Test

Pleiotropy is defined as one QTL regulates multiple traits. To take advantage of the correlation structure among multiple traits, Jiang and Zeng [24] developed a joint interval mapping method as the following equation:

$$\begin{pmatrix} y_{11} \cdots y_{1n} \\ y_{21} \cdots y_{2n} \end{pmatrix} = \begin{pmatrix} \mu_{11} \cdots \mu_{1n} \\ \mu_{21} \cdots \mu_{2n} \end{pmatrix} + \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} (x_1 \cdots x_n) + \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} (z_1 \cdots z_n) + \begin{pmatrix} e_{11} \cdots e_{1n} \\ e_{21} \cdots e_{2n} \end{pmatrix},$$

where y_i is the vector of trait values for individual i $(i = 1, \ldots, n)$, a_j and d_j are the additive and dominance effects for trait j $(j = 1, 2)$, x_j and z_i are genotypes at the test position, and e_j is the residual effect for trait j. From this statistical model a series of tests of hypotheses can be performed to test whether the two traits are supported as being driven by a single QTL at a given test position. The first test involves testing whether a given region is linked to the joint trait vector for the traits under study:

$$H_0 : a_1 = 0, d_1 = 0, a_2 = 0, d_2 = 0$$
$$H_1 : \text{at least one of the above terms is not } 0.$$

To test the above null hypothesis of no linkage against the alternative linkage hypothesis, likelihoods associated with the null and alternative hypothesis are maximized with respect to the model parameters. From the maximum likelihoods the log likelihood ratio statistic is formed and used to test whether the alternative hypothesis (H_1) is supported by the data. With this model, the log likelihood ratio statistic under the null hypothesis is chi-square distributed with 4 degrees of freedom. If the null hypothesis (H_0) is rejected, the implication is that trait 1 and/or trait 2 have a QTL at the given test locus.

Subsequent to the test just described resulting in a rejection of the null hypothesis, second and third tests of hypotheses are performed to establish whether the detected QTL affects both traits. For a given QTL test position,

$$H_{10} : a_1 = 0, d_1 = 0, a_2 \neq 0, d_2 \neq 0$$
$$H_{11} : a_1 \neq 0, d_1 \neq 0, a_2 \neq 0, d_2 \neq 0$$

assesses whether the first trait has a QTL at the test position, and

$$H_{20} : a_1 = 0, d_1 \neq 0, a_2 = 0, d_2 = 0$$
$$H_{21} : a_1 \neq 0, d_1 \neq 0, a_2 \neq 0, d_2 \neq 0$$

assesses whether the second trait has a QTL at the test position. As above, the log likelihood ratio statistics are formed for each of these tests, where under the null hypotheses these statistics are chi-square distributed with 2 degrees of freedom. If both null hypotheses H_{10} and H_{20} are rejected, the QTL is supported as having pleiotropic effects on the two traits under study.

Gene expression is noisy. Many factors can drive expression levels of two genes to be correlated [25]. In the setting of genetics crosses, genotype is the only randomized factor that drives correlations among traits. The pleiotropy test is to check whether such correlations are due to common genetic variations or other things. Figure 23.9 shows that pairs of genes correlated due to pleiotropic effects of QTLs are more coherent with regard to biological processes than pairs of genes correlated due to other factors [10].

23.3.2.3 A Likelihood-Based Causal Model Selection

We have previously published a method to infer whether two quantitative traits linked to a common genetic locus are related, with respect to the locus, in a causal, reactive or independent fashion has been previously published and validated [10,11], and extended and generalized by others [26]. To briefly review the method, for two quantitative traits T_1 and T_2 linked to the same locus L in an F2 intercross

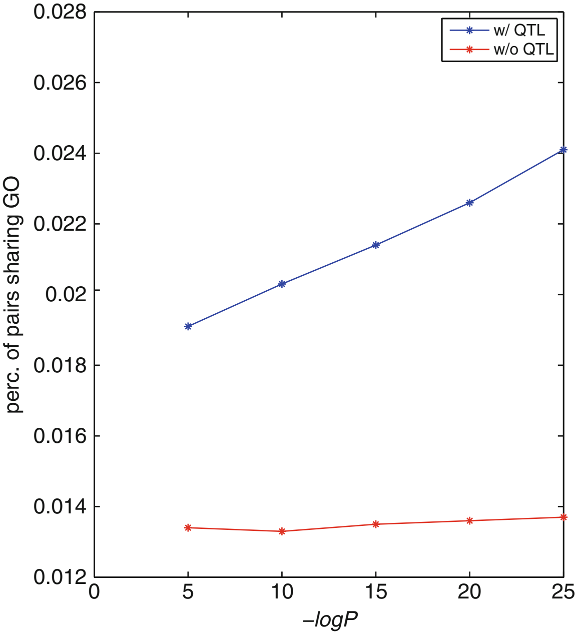


Fig. 23.9 Comparison of correlation with and without pleiotropy test filtering (QTL overlap). All pairwise correlations of gene expression of BXH ApoE male adipose are calculated. Each pair of genes are checked whether they belong to the same GO biological process. Using different correlation p-value cutoffs, there is always a higher percentage of gene pairs sharing common GO biological process after pleiotropy test filtering than without the filtering step

population, there are three basic relationships that are possible between the two traits relative to the DNA locus L. Either DNA variations at the locus L lead to changes in trait T_1 that in turn lead to changes in trait T_2, or variations at locus L lead to changes in trait T_2 that in turn lead to changes in trait T_1, or variations at locus L independently lead to changes in traits T_1 and T_2, as previously described [11]. Assuming standard Markov properties for these basic relationships, the joint probability distributions corresponding to these three models, respectively, are:

$$\begin{aligned} P\left(L,T_1,T_2\right) &= P\left(L\right) P\left(T_1|L\right) P\left(T_2|T_1\right) \\ P\left(L,T_1,T_2\right) &= P\left(L\right) P\left(T_2|L\right) P\left(T_1|T_2\right) \\ P\left(L,T_1,T_2\right) &= P\left(L\right) P\left(T_2|L\right) P\left(T_1|T_2,L\right), \end{aligned}$$

where the final term on the right-hand side of equation M3 reflects that the correlation between T_1 and T_2 may be explained by other shared loci or common environmental influences, in addition to locus L. $P\left(L\right)$ is the genotype probability distribution for locus L and is based on a previously described recombination model [24]. The random variables T_1 and T_2 are taken to be normally distributed about each genotypic mean at the common locus L, so that the likelihoods corresponding to each of the joint probability distributions are then estimated based on the normal probability density function. Because the number of model parameters among the models differs, the Bayesian Information Criteria (BIC) or Akaike Information Criterion (AIC) can be used for model selection.

To assess whether the best fitting model was significantly better than the alternative models, we developed a confidence measure using resampling methods to assess more formally whether a particular gene expression trait was causal, reactive or independent of the metabolic traits of interest, with respect to a given locus. To compute the confidence measure, 1,000 bootstrap samples were drawn. For each resample, the model selection procedure was carried out to estimate the proportion of times each model was chosen. This proportion was then considered as a reliability score for the selected model and is used to generate causal priors for Bayesian networks.

23.3.2.4 Structure Priors Derived from Genetic Data

In segregating populations, variations in DNA are the ultimate cause of traits under genetic control. An expression trait that gives rise to a cis-acting eQTL corresponds to a structural gene that harbors a DNA variant in the gene region that affects transcript levels. In constructing a network, genes with cis-acting eQTLs can be allowed to serve as parent nodes of genes with trans-acting eQTLs, $p(X_{cis} \rightarrow X_{trans}) = 1$, whereas genes with trans-acting eQTLs can not be parents of genes with cis-acting eQTLs, $p(X_{trans} \rightarrow X_{cis}) = 0$. Thus, genes with cis-acting eQTLs represent the top layers of the network.

We can further extend this concept by leveraging the genetic architecture more generally. If two genes, X_a and X_b, are found to be driven by common genetic loci (the loci have pleiotropy effect on both genes), the gene pair and the corresponding locus can be used to infer a causal/reactive or independent relationship based on a formal causality test [11] described above. The reliability of each possible relationship between gene X_a and gene X_b at locus l_i, $p(X_a \rightarrow X_b|l_i)$, $p(X_b \rightarrow X_a|l_i)$, and $p(X_a \perp X_b|l_i)$, are estimated by a standard bootstrapping procedure. If an independent relationship is inferred ($p(X_a \perp X_b|l_i) > 0.5$), then the prior probability that gene A is a parent of gene B is scaled as

$$p(X_a \rightarrow X_b) = 1 - \frac{\sum\limits_{i \in PE} p(X_a \perp X_b|l_i)}{\sum\limits_{i \in PE} 1},$$

by considering all loci where the two genes are detected as having pleiotropic effects. If a causal or reactive relationship is inferred ($p(X_a \rightarrow X_b|l_i)$ or $p(X_b \rightarrow X_a|l_i)$ is > 0.5) then the prior probability is scaled as

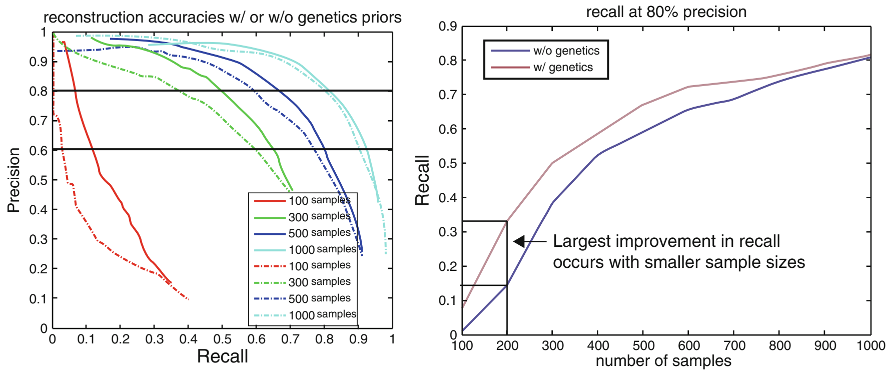
$$p(X_a \rightarrow X_b) = \frac{2 * \sum\limits_{i} p(X_a \rightarrow X_b|l_i)}{\sum\limits_{i} p(X_a \rightarrow X_b|l_i) + p(X_b \rightarrow X_a|l_i)}.$$

If the causal/reactive relationship between genes X_a and X_b can not be reasonably inferred, then the complexity of the eQTL signature for each gene can be taken into consideration. Genes with a simpler, albeit stronger eQTL signature (i.e., a small number of eQTL explain the genetic variance component for the gene, with a significant proportion of the overall variance explained by the genetic effects) can be taken to be more likely to be causal compared to genes with a more complex, possibly weaker eQTL signatures (i.e., a larger number of eQTL explaining the genetic variance component for the gene, with less of the overall variance explained by the genetic effects). In this case, the structure prior that gene X_a is a parent of gene X_b can be taken as $p(X_a \to X_b) = 2 * \frac{1+n(X_b)}{2+n(X_a)+n(X_b)}$, where $n(X_a)$ and $n(X_b)$ are the number of eQTLs with LOD scores greater than a threshold for X_a and X_b, respectively.

The improvement in the network reconstruction accuracy by incorporation of the genetic priors is shown by comparing network predicted signatures with experiment signatures [27] and by simulation studies [28]. Results showed that structure priors derived from genetic data not only help recover true causal relationship, but also help recover true relationships regardless of edge direction. The largest network reconstruction accuracy improvement due to genetic prior occurs when around 200 samples are available for network study, as shown in Fig. 23.10.

23.3.2.5 Structure Priors Derived from Other Data Sources

There are many high throughput data providing additional information to gene expression data, including protein-protein interaction data, protein-DNA binding data, microRNA binding data, and so on. When high confidence protein-protein interactions are identified, the corresponding pair of genes tend to correlate at the expression level. It is still a challenge to get high quality protein-protein interaction



Largest improvement in recall occurs with smaller sample sizes

Fig. 23.10 Comparison of accuracies of networks reconstructed with and without genetic priors. The improvement of network reconstruction accuracies depends on the number of samples used in network reconstruction. The largest improvement occurs when there are 200 or so samples

data and protein-DNA binding data at genome scale for mammalians. However, there is high quality transcription factor binding site data for yeast which was derived from high quality ChIP-on-Chip experiments and phylogenetic conservation filter [29], and protein-protein interaction data was derived from manually curated protein complexes [30] as well as from complexes identified by clique community analysis. These data can be combined together to form a structure prior used to reconstruct Bayesian networks. We have demonstrated that the derived priors can significantly enhance network reconstruction accuracy [31].

Gene expression is regulated by transcription factors, and many gene-gene expression correlations can be explained by co-regulation by the same transcription factors. Gene expression networks exhibit a scale-free property which is a general property of biological networks [32]. The scale-free property suggests that a small numbers of transcription factors regulated a large number of genes' expression levels. To represent TF-DNA and protein-protein complex data in the network reconstruction, we introduce a scale-free prior. For example, given a transcription factor T, and a set of genes G that contain the binding site of T, the transcription factor prior p_{tf} can be defined so that it is proportional to the number of responders that are correlated with the transcription factor's expression level $\log(p_{tf}(T \rightarrow g)) \propto \log(\sum_{g_i \in G} p_{qtl}(T \rightarrow g_i) * \delta)$, where $p_{qtl}(T \rightarrow g)$ is the structure prior for the QTL and $\delta = \begin{cases} 1, \; if corr(T, g_i) \geq r_{cutoff} \\ 0, \; if corr(T, g_i) < r_{cutoff} \end{cases}$. The correlation cutoff r_{cutoff} can be determined using permuted data to minimize the false discovery rate. When the Bayesian networks based on these yeast data were reconstructed using these priors and compared to the yeast knock-out compendium data [18], there are 125, 139 and 152 knock-out signatures enriched in the networks reconstructed using only expression data, using expression and genetic data only, and using expression, genetic, TF binding site and protein-protein interaction data, respectively. These results indicate that the integration of orthogonal experimental data improves the quality of reconstructed networks, and these more predictive networks will have greater utility in refining the definition of disease, identifying disease subtypes, identifying targets for disease, and identifying biomarkers for disease and drug response.

23.3.2.6 More Details of Mechanism Revealed by Integrating Additional Data

Orthogonal data not only provides prior information for network reconstruction, but also provides mechanism explanation of network regulation. For example, there are 13 eQTL hot spots for the yeast segregant data described in previous section [20]. LEU2 is predicted as one of regulators for eQTL hot spot on chromosome 3 [20,31]. We have shown that LEU2 knockout signature significantly overlap with LEU2 subnetwork (p-value $= 4.91 \times 10^{-18}$). Based on available high quality TF-DNA binding data, we noticed that genes with LEU3 binding sites are enriched in both LEU2

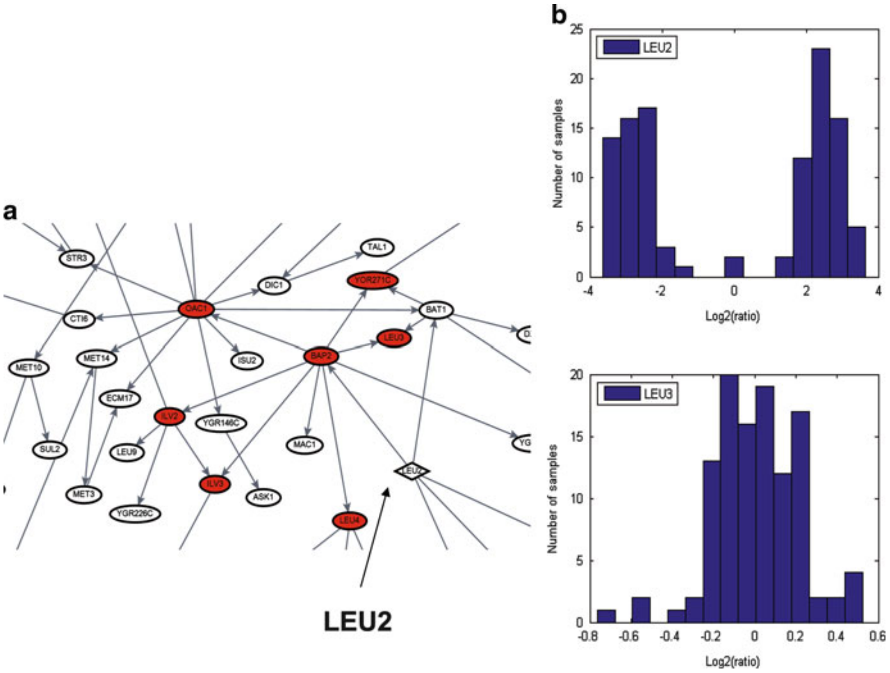


Fig. 23.11 The effect of LEU3 on LEU2 subnetwork. (**a**) Genes with LEU3 binding sites (*red* nodes) are enriched in the LEU2 subnetwork (p-value $= 1.42 \times 10^{-8}$) and close to LEU2 itself in the network. (**b**) LEU2 expression level variates among the segregant population, how LEU3 expression level does not change among the segregant population. This suggests that LEU2 expression affects LEU3 activity instead of LEU3 expression level

subnetwork and LEU2 knockout signature (p-values $= 1.42\times10^{-8}$ and 7.52×10^{-5}, respectively), shown in Fig. 23.11a. We hypothesized that the mechanism of LEU2 mutation affects the eQTL hot spot as following: LEU2 genotype affects LEU2 expression level which in turn affects LEU3, then LEU3 regulates genes with LEU3 binding sites. However, LEU3 transcriptional level does not change among the segregant population, shown in Fig. 23.11b. There is a missing link how LEU2 expression affects LEU3 activity. To answer this question, we quantified metabolites for the segregants using quantitative NMR. The concentration of an intermediate metabolite in leucine biosynthesis reactions, 2-isoproprylmalate, is linked to the eQTL hot spot on Chromosome 3, shown in Fig. 23.12a. LEU2 expression level causally affects 2-isopropylmalate concentration, and 2-isopropylmalate is causal for expression of genes with LEU3 binding sites, shown in Fig. 23.12b. It has been shown that 2-isopropylmalte binds LEU3 protein regulates target gene activation by LEU3 [33]. Then, the mechanism of LEU2 affecting genes linked the eQTL hot spot is clearer: LEU2 genotype affects LEU2 expression level which in turn affects 2-isopropylmalate concentration, then 2-isopropylmalte binds to LEU3 protein and regulates LEU3 protein activity, LEU3 regulates genes with LEU3 binding sites.

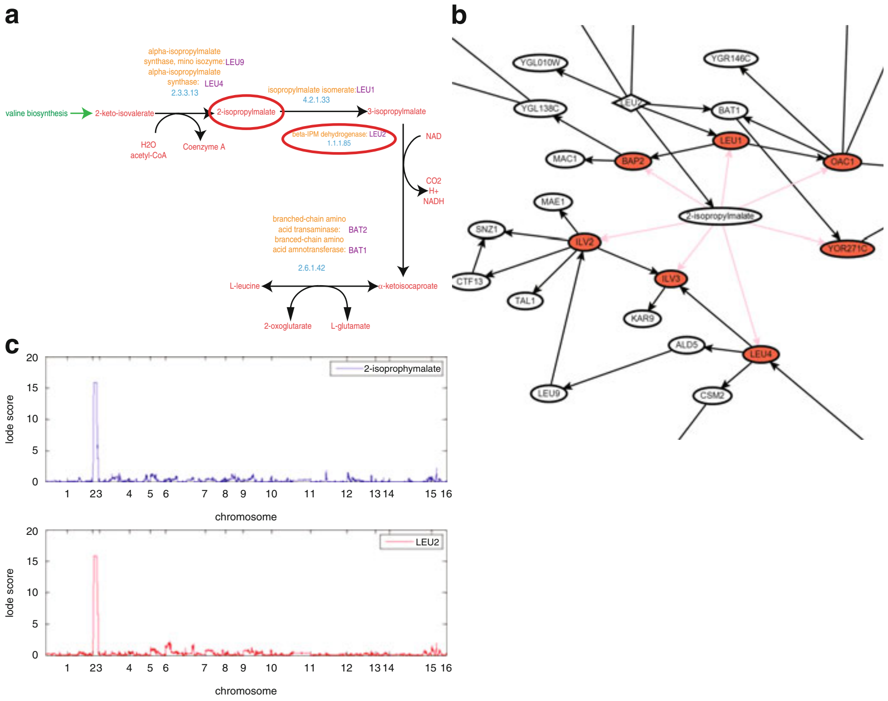


Fig. 23.12 Relationship of LEU2 and 2-isopropylmalate. (**a**) Both LEU2 and 2-isopropylmalate are key components in KEGG leucine biosynthesis pathway. (**b**) LEU2 expression and 2-isopropylmalate concentration are linked to LEU2 locus and LEU2 expression is causal to 2-isopropylmalate based on the causality test. (**c**) 2-isopropylmalate is directly causal to genes with LEU3 binding sites (*red* nodes). It has been shown that 2-isopropylmalate modulates Leu3p activity [33]

23.4 Related Approaches

Although we just focus on two selected approaches as examples to illustrate systems methods of studying biological networks, other approaches are emerging too. For example, a recently developed approach built on Tu et al.'s method, called eQED, has been demonstrated to improve the performance of predicting the regulator-target relationship [34]. In this method, analog electric circuit is used to model the protein interaction and gene regulatory network. The weights on the edges of the molecular network, defined as the average of the mRNA correlation of genes associated with the edge with the target gene, are used to represent conductance in the circuit. By putting some voltage on the circuit and assuming the target gene is grounded, the causal gene in the eQTL is predicted as the one with the highest current running through it. This model is very similar to Tu et al.'s method except that branches with dead-end are not counted and are excluded from calculation. By doing so, Suthram et al. showed ∼50% increase in the number of correct predictions of causal gene-target pairs by simple eQED model or $\sim$67% increase when multiple loci are considered simultaneously using a globally constructed circuit.

There are also several regression based methods to predict causal/reactive relationship assuming that the pool of all possible causal regulators is known [35–37]. One of regression based methods, called Lirnet, has been shown to generate better prediction accuracy [37]. In most of these methods, genetic data or eQTL is the key component for accurately inferring causal relationship. Genes predicted to be causal to obesity by Schadt et al. [11] have been systematically tested and validated [38]. Many variants of the causality test based on eQTL have been proposed [15, 26, 39]. Causal prediction confidence estimated by a series of permutation tests is shown to be more accurate than the bootstrapping method [40]. In addition to these methods centered on genetic regulation of gene expression, there are significantly more methods existing when considering integrative systems approaches in a more general sense [24, 41, 42].

23.5 Conclusions

Unwiring the signaling and regulatory network is of great importance for understanding the biological system, which in turn will greatly help us to reveal the underlying mechanism of diseases and facilitate developing novel treatment. However, as we have pointed out, it is highly challenging to untangle the complex system due to the large number of genes involved as both input and output and the large number of parameters required to model signal transition from input to output. We demonstrate that integrating various types of high throughput data that capture different aspects of the system provides a promising way to tackle the problem. Although these methods are different in their designs and assumptions about how complete our knowledge is, they are common in demonstrating that only by considering multiple sources of information, and investing all these variables simultaneously, can we obtain a much clearer view of the system.

As one of the most rapidly developing area in current biology research, we expect that the integrative approach of studying signaling and regulatory network will continue to play a critical role in advancing our understanding of the biological system.

References

1. Rikova, K., Guo, A., Zeng, Q., et al. (2007). Global survey of phosphotyrosine signaling identifies oncogenic kinases in lung cancer. *Cell*, *131*(6), 1190–1203.
2. Creighton, C. J. (2008). Multiple oncogenic pathway signatures show coordinated expression atterns in human prostate tumors. *PLos ONE*, *3*(3), e1816.
3. Zaman, S., Lippman, S. I., Zhao, X., et al. (2008). How saccharomyces responds to nutrients. *Annual Review of Genetics*, *42*(1), 27–81.

4. Wang, Y., Pierce, M., Schneper, L., et al. (2004). Ras and Gpa2 mediate one branch of a redundant glucose signaling pathway in yeast. *PLos Biology*, *2*(5), e128.
5. Zaman, S., Lippman, S. I., Schneper, L., et al. (2009). Glucose regulates transcription in yeast through a network of signaling pathways. *Molecular Systems Biology*, *5*, 245.
6. Ptacek, J., Devgan, G., Michaud, G., et al. (2005). Global analysis of protein phosphorylation in yeast. *Nature*, *438*(7068), 679–684.
7. Morley, M., Molony, C. M., Weber, T. M., et al. (2004). Genetic analysis of genome-wide variation in human gene expression. *Nature*, *430*(7001), 743–747.
8. Brem, R. B., Yvert, G., Clinton, R., et al. (2002). Genetic dissection of transcriptional regulation in budding yeast. *Science*, *296*(5568), 752–755.
9. Schadt, E. E., et al. (2003). Genetics of gene expression surveyed in maize, mouse and man. *Nature*, *422*, 297–302.
10. Chen, Y., Zhu, J., Lum, P. Y., et al. (2008). Variation in DNA elucidate molecular networks that cause disease. *Nature*, *452*(7186), 429–435.
11. Schadt, E. E., et al. (2005). An integrative genomics approach to infer causal associations between gene expression and disease. *Nature Genetics*, *37*, 710–717.
12. Brem, R. B., & Kruglyak, L. (2005). The landscape of genetic complexity across 5,700 gene expression traits in yeast. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(5), 1572–1577.
13. Brem, R. B., Storey, J. D., Whittle, J., et al. (2005). Genetic interactions between polymorphisms that affect gene expression in yeast. *Nature*, *436*(7051), 701–703.
14. Rockman, M. V., & Kruglyak, L. (2006). Genetics of global gene expression. *Nature Reviews Genetics*, *7*(11), 862–872.
15. Tu, Z., Wang, L., Arbeitman, M. N., et al. (2006). An integrative approach for causal gene identification and gene regulatory pathway inferece. *Bioinformatics*, *22*(14), e489–e496.
16. Ghaemmaghami, S., Huh, W.-K., Bower, K., et al. (2003). Global analysis of protein expression in yeast. *Nature*, *425*(6959), 737–741.
17. Zien, A., Kuffner, R., Zimmer, R., et al. (2000). Analysis of gene expression data with pathway scores. *Proceedings of the International Conference on Intelligent Systems and Molecular Biology*, *8*, 407–417.
18. Hughes, T. R., Marton, M. J., Jones, A. R., et al. (2000). Functional discovery via a compendium of expression profiles. *Cell*, *102*(1), 109–126.
19. Wang, Y., & Dohlman, H. G. (2004). Pheromone signaling mechanisms in yeast. *Science*, *306*(5701), 1508–1509.
20. Yvert, G., Brem, R. B., Whittle, J., et al. (2003). Trans-acting regulatory variation in Saccaromyces cerevisiae and the role of transcription factors. *Nature Genetics*, *35*(1), 57–64.
21. Zhu, J., Zhang, B., Smith, E. N., et al. (2008). Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. *Nature Genetics*, *40*(7), 854–861.
22. Fujita, A., Sato, J. R., Garay-Malpartida, H. M., et al. (2007). Time-varying modeling of gene expression regulatory networks using the wavelet dynamic vector autoregressive method. *Bioinformatics*, *23*(13), 1623–1630.
23. Yu, J., Smith, V. A., Wang, P. P., et al. (2004). Advances to bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, *20*(18), 3594–3603.
24. Wu, X., Jiang, R., Zhang, M. Q., et al. (2008). Network-based global inference of human disease genes. *Molecular Systems Biology*, *4*, 189.
25. Stuart, J. M., Segal, E., Koller, D., et al. (2003). A gene-coexpression network for global discovery of conserved genetic modules. *Science*, *302*(5643), 249–255.
26. Start, J. M., Jagalur, M., et al. (2006). Causal inference of regulator-target pairs by gene mapping of expression phenotypes. *BMC Genomics*, *7*, 125.
27. Zhu, J., Lum, P. Y., Lamb, J., et al. (2004). An integrative genomics approach to the reconstruction of gene networks in segregating populations. *Cytogenetics and Genome Research*, *105*(2–4), 363–374.

28. Zhu, J., Wiener, M. C., Zhang, C., et al. (2007). Increasing the power to detect causal associations by combining genotypic and expression data in segregating populations. *Plos Computational Biology*, *3*(4), e69.
29. MacIsaac, K. D., Wang, T., Gordon, D. B., et al. (2006). An improved map of conserved regulatory sites for Saccharomyces cerevisiae. *BMC Bioinformatics*, *7*, 113.
30. Guldener, U., Munsterkotter, M., Oesterheld, M., et al. (2006). MPact: The MIPS protein interaction resource on yeast. *Nucleic Acids Research*, *34*, D436–441.
31. Zhu, J., Zhang, B., Smith, E. N., et al. (2008). Integrating large-scale functional genomic data to dissect the complexity of yeast regulatory networks. *Nature Genetics*, *40*(7), 854–861.
32. Albert, R., Jeong, H., Barabasi, A. L., et al. (2000). Error and attack tolerance of complex networks. *Nature*, *406*(6794), 378–382.
33. Sze, J. Y., Woontner, M., Jaehning, J. A., et al. (1992). In vitro transcriptional activation by a metabolic intermediate: Activation by Leu3 depends on alpha-isopropylmalate. *Science*, *258*(5085), 1143–1145.
34. Suthram, S., Beyer, A., Karp, R. M., et al. (2008). eQED: An efficient method for interpreting eQTL associations using protein networks. *Molecular Systems Biology*, *4*, 162.
35. Basso, K., Margolin, A. A., Stolovitzky, G., et al. (2005). Reverse engineering of regulatory networks in human B cells. *Nature Genetics*, *37*,(4), 382–390.
36. Lee, S. I., Pe'er, D., Dudley, A. M., et al. (2006). Identifying regulatory mechanisms using individual variation reveals key role for chromatin modification. *Proceedings of the National Academy of Sciences of the United States of America*, *103*(38), 14062–14067.
37. Lee, S. I., Dudley, A. M., Drubin, D., et al. (2009). Learning a prior on regulatory potential from eQTL data. *PLoS Genetics*, *5*(1), e1000358.
38. Yang, X., Deignan, J. L., Qi, H., et al. (2009). Validation of candidate causal genes for obesity that affect shared metabolic pathways and networks. *Nature Genetics*, *41*(4), 415–423.
39. Chaibub Neto, E., Ferrara, C. T., Attie, A. D., et al. (2008). Inferring causal phenotype networks from segregating populations. *Genetics*, *179*(2), 1089–1100.
40. Chen, L. S., Emmert-Streib, F., & Storey, J. D. (2007). Harnessing naturally randomized transcription to infer regulatory relationships among genes. *Genome Biology*, *8*(10), R219.
41. Cui, Q., Ma, Y., Jaramillo, M., et al. (2007). A map of human cancer signaling. *Molecular Systems Biology*, *3*, 152.
42. Chuang, H.-Y., Lee, E., Liu, Y.-T., et al. (2007). Network-based classification of breast cancer metastasis. *Molecular Systems Biology*, *3*, 140.

Chapter 24
Computational Drug Target Pathway Discovery: A Bayesian Network Approach

Seiya Imoto, Yoshinori Tamada, Hiromitsu Araki, and Satoru Miyano

Abstract Genome-wide transcriptome data together with statistical analysis enable us to reverse-engineer gene networks that can be a kind of views useful for understanding dynamic behaviour of biological elements in cells. In this chapter, we elucidate statistical models for estimating gene networks based on two types of microarray gene expression data, gene knock-down and time-course. In our modeling, nonparametric regression model is combined with Bayesian networks to capture nonlinear relationships between genes and a derived Bayesian information criterion with efficient structure learning algorithm selects network structure. Some efficient algorithms for structure learning of Bayesian networks, which is known as an NP-hard problem for optimal solutions, are also introduced. To demonstrate the statistical gene network analysis shown in this chapter, we estimate gene networks based on microarray data of human endothelial cell treated with an anti-hyperlipidaemia drug fenofibrate. Based on the constructed gene networks, we illustrate computational strategies for discovering drug target genes and pathways.

S. Imoto
Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo, 108-8639 Japan
e-mail: imoto@ims.u-tokyo.ac.jp

Y. Tamada
Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo, 108-8639 Japan
e-mail: tamada@ims.u-tokyo.ac.jp

H. Araki
Cell Innovator Inc.
e-mail: hiromitsu_araki@cell-innovator.com

S. Miyano (✉)
Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo, 108-8639 Japan
e-mail: miyano@ims.u-tokyo.ac.jp

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_24,
© Springer-Verlag Berlin Heidelberg 2011

24.1 Introduction

Due to the advances of microarray technology, large amount of gene expression data have been measured in various species, human cell lines, disease cells, cells with various stimuli and so on. Construction of gene networks that enables us to understand relationships between biological molecules in genome-wide scale is considered as a challenging problem but absolutely essential for systems biology. Several mathematical models, including Boolean networks [3–5, 70], Bayesian networks [18, 27, 35, 37, 70], graphical Gaussian models [68, 76], dynamic Bayesian networks [46,52,55], vector autoregressive models [19], state space models [32,79], etc., have been proposed for reverse-engineering gene networks based on microarray data.

In this chapter, we describe statistical methods based on Bayesian networks for constructing gene networks. Bayesian networks have been developed mainly in the field of artificial intelligence as an expert system. The theory and methodology for Bayesian network learning have been well studied for discrete data. The gene expression values are, however, essentially continuous and we need to convert them into discrete values, if we wish to use the discrete-type Bayesian networks. However, the discretization leads to information loss and the threshold values for discretization are also problematic parameters. Moreover, the number of categories in the discretization should be chosen appropriately. The resulting networks strongly depend on their values. One possible research direction might be to study these problems in microarray data in order to use the discrete-type Bayesian networks. Another and important direction is, however, to extend Bayesian networks suitable for analysis of continuous microarray data. To use microarray data as continuous variables, Friedman et al. [18] considered fitting linear regression models (see also Heckerman and Geiger [31]). However, the assumption that the parent genes depend linearly on the objective gene is not always guaranteed. Imoto et al. [35, 37] then proposed the use of nonparametric additive regression models (see also Green and Silverman [23] and Hastie and Tibshirani [28]) for capturing not only linear dependencies but also nonlinear relationships between genes.

Although modeling of the relationship between genes is one of the important tasks in gene network estimation, a more fundamental issue is how we determine the structure of gene network. This problem is equivalent to the structure learning of Bayesian networks. In this chapter, we consider structure learning of Bayesian networks from a statistical model evaluation point of view. More concretely, we choose an optimal Bayesian network structure based on the statistical model evaluation using an information criterion. Based on Bayesian statistics, we derive an information criterion termed BNRC. BNRC is considered as an extension of Bayesian information criterion proposed by Schwarz [62], which is used to evaluate models estimated by maximum likelihood method, to evaluate models estimated by maximum penalized likelihood method.

To improve the quality of the estimated gene networks, we also introduce a way to combine biological prior knowledge with microarray data to estimate gene networks. For example, the information of cis-regulatory elements can be used.

A Bayesian framework is provided for this purpose; the microarray data is used to build the likelihood based on Bayesian networks and the biological prior knowledge is used for constructing prior probability of the network structure.

Computational challenges using gene network estimation technology are to uncover the mode-of-action of a drug and novel drug target pathways. We show a strategic method for this challenge by analyzing drug response time-course microarray data and gene knock-down microarray data with extended Bayesian networks with the application of microarray data of human endothelial cell treated with an anti-hyperlipidaemia drug, fenofibrate.

24.2 Statistical Modelings for Gene Networks

24.2.1 Bayesian Networks and Nonparametric Regression

24.2.1.1 Bayesian Networks

Bayesian network is a probabilistic graphical model that gives a compact representation of joint probability of a large number of random variables. Let $\mathscr{X} = \{X_1, \ldots, X_p\}$ be a set of random variables and let G be a directed acyclic graph that represents statistical or causal dependency among $X_1, \ldots, X_p$. Mathematically, a random variable X_j is regarded as a node of G and a directed acyclic graph G is defined by $G = (\mathscr{X}, \mathscr{E})$, where $\mathscr{E}$ is the set of direct edges; a direct edge $e(i, j)$ is included in $\mathscr{E}$ if and only of there exits the direct edge from X_i to X_j in G.

By assuming the Markov property between nodes in G, i.e., a node depends only on its direct parents and independent of other non-descendant nodes, the joint probability of $X_1, \ldots, X_p$ can be decomposed as:

$$\Pr(X_1, \ldots, X_p) = \prod_{j=1}^{p} \Pr(X_j | Pa(X_j)), \tag{24.1}$$

where $Pa(X_j)$ is the set of the direct parents of X_j in G. We note that Eq. 24.1 specifies conditional independencies among random variables $X_1, \ldots, X_p$, e.g., X_j and random variables in $ND(X_j) \setminus Pa(X_j)$ are conditionally independent when $Pa(X_j)$ is given, where $ND(X_j)$ is the set of non-descendant random variables of X_j; we obtain $\Pr(X_j|ND(X_j)) = \Pr(X_j|Pa(X_j))$.

In Friedman et al. [18], Hartemink et al. [26] and Pe'er et al. [58], microarray gene expression values were discretized into several categorical values, e.g., $c_1, c_2, \ldots$, and they used discrete-type Bayesian networks specified by a probability table for modeling gene networks:

$$\theta_{ijk} = \Pr(X_i = u_{ij} | Pa(X_i) = \mathbf{u}_{ik}),$$

where u_{ij} is the value corresponding to jth category and $\mathbf{u}_{ik}$ is the kth pattern of the parent nodes of X_i. Typically, for the data discretization, one can set three categorical values $u_{i1} = -1$, $u_{i2} = 0$ and $u_{i3} = 1$, where $X_i = -1, 0$ or 1 represent ith gene is repressed, unchanged or overexpressed, respectively [18]. As we mentioned above, the use of discrete-type Bayesian networks for microarray data analysis has some problems. In the next section, we introduce a nonparametric regression for extending Bayesian networks to estimate gene networks from microarray data without discretization.

24.2.1.2 Nonparametric Regression

Friedman et al. [18] considered fitting linear regression models, which analyze the microarray gene expression data as continuous variables (see also Heckerman and Geiger [31]). Suppose that we have the observational data $\mathbf{X}_n$ of the set of p random variables $\mathscr{X} = \{X_1, \ldots, X_p\}$, where $\mathbf{X}_n$ is an $(n \times p)$ matrix whose (i, j)th element, x_{ij}, corresponds to the expression value of jth gene measured by ith microarray. In this content, a gene is regarded as a random variable representing the abundance of a specific RNA species. A linear regression model for jth gene can be represented by $x_{ij} = \beta_{0j} + \sum_{k:X_k \in Pa(X_j)} \beta_{kj} x_{ik} + \varepsilon_{ij}, \;\; (i = 1, \ldots, n)$, where β_{0j} and β_{kj}'s are parameters, and ε_{ij}'s are noise terms independently generated from identical distribution with zero mean and finite variance. Usually, one can use Gaussian distribution for ε_{ij} that yields Gaussian linear regression model.

In linear regression models described above, the assumption that the parent genes depend linearly on the objective gene is not always guaranteed. Then Imoto et al. [35, 37] proposed the use of nonparametric additive regression models (see also Green and Silverman [23] and Hastie and Tibshirani [28]) for capturing not only linear dependencies but also nonlinear relationships between genes. In general, nonparametric regression with additive noise can be represented by

$$x_{ij} = m_j(\mathbf{pa}(X_j)_i) + \varepsilon_{ij}, \;\; (i = 1, \ldots, n), \tag{24.2}$$

where $\mathbf{pa}(X_j)_i$ is the vector of expression values of the parents of X_j measured by ith microarray and $m(\cdot)$ is a smooth function. The additive assumption for the regressor can yield

$$m_j(\mathbf{pa}(X_j)_i) = \sum_{k:X_k \in Pa(X_j)} m_{jk}(x_{ik}), \tag{24.3}$$

where $m_{jk}(\cdot)$ is a smooth function. We construct m_{jk} by a basis expansion approach

$$m_{jk}(x_{ik}) = \sum_{\alpha=1}^{M_{jk}} \gamma_{\alpha jk} b_{\alpha jk}(x_{ik}), \tag{24.4}$$

where $\{b_{1jk}(\cdot), \ldots, b_{M_{jk}jk}(\cdot)\}$ is the prescribed set of basis functions, $\gamma_{\alpha jk}$'s are parameters and M_{jk} is the number of basis functions.

For continuous data, the decomposition formula for joint probability in Eq. 24.1 can be rewritten as

$$f(x_{i1}, \ldots, x_{ip}|G) = \prod_{j=1}^{p} f_j(x_{ij}|\mathbf{pa}(X_j)_i), \quad (i = 1, \ldots, n), \tag{24.5}$$

where f and f_j's are densities. Hence, when we use Gaussian noise, i.e., $\varepsilon_{ij} \sim N(0, \sigma_j^2)$, a statistical model for gene networks based on Bayesian network and nonparametric regression is given by

$$\begin{aligned} & f_j(x_{ij}|\mathbf{pa}(X_j)_i, \boldsymbol{\theta}_j) \\ &= \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left[-\frac{\{x_{ij} - \sum_{k:X_k \in Pa(X_j)} \sum_{\alpha=1}^{M_{jk}} \gamma_{\alpha jk} b_{\alpha jk}(x_{ik})\}^2}{2\sigma_j^2} \right], \end{aligned} \tag{24.6}$$

where $\boldsymbol{\theta}_j$ is the vector of parameters in f_j, e.g., $\boldsymbol{\theta}_j = (\gamma_{1j1}, \ldots, \sigma_j^2)'$.

24.2.2 *Dynamic Bayesian Networks and Nonparametric Regression*

24.2.2.1 Dynamic Bayesian Networks

Dynamic Bayesian network is an extension of the Bayesian network for analyzing time-course data. Let X_{tj} be a random variable governing the expression value of jth gene at time t. Put $\mathscr{X}_t = \{X_{t1}, \ldots, X_{tp}\}$ and $\mathscr{X}_{1:T} = \{\mathscr{X}_1, \ldots, \mathscr{X}_T\}$. Like Bayesian network, we set the Markov property between $\mathscr{X}_t$ and $\mathscr{X}_{t-1}$ as

$$\Pr(\mathscr{X}_t|\mathscr{X}_{t-1}, \mathscr{X}_{t-2}, \ldots, \mathscr{X}_1) = \Pr(\mathscr{X}_t|\mathscr{X}_{t-1}).$$

In each time slice, $\Pr(\mathscr{X}_t|\mathscr{X}_{t-1})$, we construct a network representing gene regulations. The network structure is assumed to be stable throughout all time points. Taking these gene regulations, the conditional probability $\Pr(\mathscr{X}_t|\mathscr{X}_{t-1})$ can also be decomposed into the product of conditional probabilities of each gene given its parent genes, of the form

$$\Pr(\mathscr{X}_t|\mathscr{X}_{t-1}) = \prod_{j=1}^{p} \Pr(X_{tj}|Pa(X_j)_{t-1}),$$

where $Pa(X_j)_{t-1}$ is the state vector of the parent genes of jth gene at time $t-1$. Hence, like the Bayesian network, we obtain the following decomposition of joint probability:

$$\Pr(\mathscr{X}_{1:T}) = \Pr(\mathscr{X}_1)\prod_{t=2}^{T}\prod_{j=1}^{p}\Pr(X_{tj}|Pa(X_j)_{t-1}).$$

Therefore, an essential point of modeling dynamic Bayesian network is to construct the conditional probability $\Pr(X_{tj}|Pa(X_j)_{t-1})$.

24.2.2.2 Nonparametric Vector Auto-Regression

For microarray time-course data, we focus on the construction of the density $f_j(x_{tj}|\mathbf{pa}(X_j)_{t-1})$. Like nonparametric regression model for Bayesian networks, we can extend the nonparametric regression model in Eq. 24.2 into the first-order nonparametric autoregressive model of the form

$$x_{tj} = m_j(\mathbf{pa}(X_j)_{t-1}) + \varepsilon_{tj}, \quad (t = 2, \ldots, T),$$

with additive regressor defined by combining Eqs. 24.3 and 24.4. This model is considered as an extension of linear autoregressive model defined by $m_j(\mathbf{pa}(X_j)_{t-1}) = \boldsymbol{\beta}'_j\mathbf{pa}(X_j)_{t-1}$, where $\boldsymbol{\beta}_j$ is the vector of coefficients and $\boldsymbol{\beta}'_j$ indicates the transpose of the vector $\boldsymbol{\beta}_j$. Therefore, the dependencies detected by this model are considered as a nonlinear Granger's causality [22].

24.2.3 Statistical Model Selection Approach for Learning Bayesian Networks

24.2.3.1 Parameter Estimation

The Bayesian network and nonparametric regression model defined by combining Eqs. 24.5 and 24.6 has parameters of coefficients for basis functions and variances of noise. From a Bayes approach, given a graph G, maximum a posteriori estimate of the parameter $\boldsymbol{\theta}_j$ are defined by

$$\hat{\theta}_j = \arg\max_{\boldsymbol{\theta}_j}\prod_{i=1}^{n} f_j(x_{ij}|\mathbf{pa}(X_j)_i, \boldsymbol{\theta}_j)\pi_j(\boldsymbol{\theta}_j|\boldsymbol{\lambda}_j), \tag{24.7}$$

where $\pi_j(\boldsymbol{\theta}_j|\boldsymbol{\lambda}_j)$ is the prior distribution on the parameter $\boldsymbol{\theta}_j$ with the hyperparameter vector $\boldsymbol{\lambda}_j$. Suppose that the prior distribution $\pi_j(\boldsymbol{\theta}_j|\boldsymbol{\lambda}_j)$ is factorized as

$$\pi_j(\boldsymbol{\theta}_j|\boldsymbol{\lambda}_j) = \prod_{k:X_k \in Pa(X_j)} \pi_{jk}(\boldsymbol{\gamma}_{jk}|\lambda_{jk}),$$

where $\boldsymbol{\gamma}_{jk} = (\gamma_{1jk}, \dots, \gamma_{M_{jk}jk})'$ and λ_{jk}'s are hyperparameters that control the preciseness of the prior knowledge. In practice, we use a singular M_{jk} variate normal distribution as the prior distribution on $\boldsymbol{\gamma}_{jk}$,

$$\pi_{jk}(\boldsymbol{\gamma}_{jk}|\lambda_{jk}) = \left(\frac{2\pi}{n\lambda_{jk}}\right)^{-(M_{jk}-2)/2} |K_{jk}|_+^{1/2} \exp\left(-\frac{n\lambda_{jk}}{2}\boldsymbol{\gamma}'_{jk}K_{jk}\boldsymbol{\gamma}_{jk}\right), \tag{24.8}$$

where K_{jk} is an $M_{jk} \times M_{jk}$ symmetric positive semidefinite matrix satisfying

$$\boldsymbol{\gamma}'_{jk}K_{jk}\boldsymbol{\gamma}_{jk} = \sum_{\alpha=3}^{M_{jk}} (\gamma_{\alpha jk} - 2\gamma_{\alpha-1jk} + \gamma_{\alpha-2jk})^2.$$

By taking the logarithm of the objective function in Eq. 24.7, we immediately find the MAP estimate of $\boldsymbol{\theta}_j$ is equivalent to the maximum penalized log-likelihood that, by taking the minus and omitting terms independent of parameters, is the solution of the following optimization:

$$\hat{\boldsymbol{\theta}}_j = \arg\min_{\boldsymbol{\theta}_j} \left[\log(\sigma_j^2) + \frac{1}{\sigma_j^2} \sum_{i=1}^{n} \left\{ x_{ij} - \sum_{k:X_k \in Pa(X_j)} \sum_{\alpha=1}^{M_{jk}} \gamma_{\alpha jk} b_{\alpha jk}(x_{ik}) \right\}^2 \right.$$
$$\left. -n \sum_{k:X_k \in Pa(X_j)} \lambda_{jk} \boldsymbol{\gamma}'_{jk} K_{jk} \boldsymbol{\gamma}_{jk} \right].$$

Put $\mathbf{x}_{(j)} = (x_{1j}, \dots, x_{nj})'$ and $B_{jk} = (\mathbf{b}_{jk}(x_{1k}), \dots, \mathbf{b}_{jk}(x_{nk}))'$ with $\mathbf{b}_{jk}(x_{ik}) = (b_{1jk}(x_{ik}), \dots, b_{M_{jk}jk}(x_{ik}))'$, $|Pa(X_j)| = q_j$ and $X_{j_m} \in Pa(X_j)$ $(m = 1, \dots, q_j)$. Based on the backfitting algorithm [28], the modes $\hat{\boldsymbol{\gamma}}_{jk}$ can be obtained by the following procedure:

Step 1 Initialize: $\boldsymbol{\gamma}_{jk} = \mathbf{0}$, for all k such that $X_k \in Pa(X_j)$.
Step 2 Cycle: $k = j_1, \dots, j_{q_j}, j_1, \dots, j_{q_j}, j_1, \dots$

$$\boldsymbol{\gamma}_{jk} = (B'_{jk}B_{jk} + n\beta_{jk}K_{jk})^{-1}B'_{jk}\left(\mathbf{x}_{(j)} - \sum_{l \neq k} B_{jl}\boldsymbol{\gamma}_{jl}\right), \tag{24.9}$$

where β_{jk} is equivalent to $\hat{\sigma}_j^2\lambda_{jk}$ and set to a fixed value.
Step 3 Continue Step 2 until a suitable convergence criterion is satisfied.

The mode $\hat{\sigma}_j^2$ is given by $\hat{\sigma}_j^2 = ||\mathbf{x}_{(j)} - \sum_{k:X_k \in Pa(X_j)} B_{jk}\hat{\boldsymbol{\gamma}}_{jk}||^2/n$ and $\hat{\boldsymbol{\gamma}}_{jk}$ is the final updated. We note that the estimates $\hat{\boldsymbol{\gamma}}_{jk}$ and $\hat{\sigma}_j^2$ depend on the values of the hyperparameters β_{jk}. The hyperparameters β_{jk} are called smoothing parameters in the context of nonparametric regression. There are many methods for choosing smoothing parameters such as cross-validation, generalized cross validation, Akaike's information criterion [2], Bayesian information criterion [62] and so on. In our context, we choose the hyperparameters based on an information criterion derived in the next section.

24.2.3.2 Statistical Evaluation for Network Structure

Suppose that we have microarray data $\mathbf{X}_n$ of the set of p genes $\mathscr{X} = \{X_1, \ldots, X_p\}$ and that the dependency among p genes, shown as a directed graph G, is unknown and we want to estimate it from $\mathbf{X}_n$. From a Bayes approach, the optimal graph is selected by maximizing the posterior probability of the graph conditional on the observed data. By Bayes' theorem, the posterior probability of the graph can be represented as

$$p(G|\mathbf{X}_n) = \frac{p(G)p(\mathbf{X}_n|G)}{p(\mathbf{X}_n)} \propto p(G)p(\mathbf{X}_n|G), \tag{24.10}$$

where $p(G)$ is the prior probability of the graph, $p(\mathbf{X}_n|G)$ is the likelihood of the data $\mathbf{X}_n$ conditional on G and $p(\mathbf{X}_n)$ is the normalizing constant and does not depend on the selection of G. Therefore, we need to set $p(G)$ and compute $p(\mathbf{X}_n|G)$ for the graph selection based on $p(G|\mathbf{X}_n)$.

The likelihood $p(\mathbf{X}_n|G)$ can be computed by Bayesian networks or dynamic Bayesian networks. By removing the normalizing constant, the likelihood of the data is given by

$$p(\mathbf{X}_n|G) = \int \prod_{i=1}^{n} f(\mathbf{x}_i|\boldsymbol{\theta}_G)\pi(\boldsymbol{\theta}_G|\boldsymbol{\lambda})d\boldsymbol{\theta}_G, \tag{24.11}$$

where $\boldsymbol{\theta}_G = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p)'$ and $\pi(\boldsymbol{\theta}_G|\boldsymbol{\lambda})$ is the density of the prior distribution on $\boldsymbol{\theta}_G$ with the hyperparameter vector $\boldsymbol{\lambda}$. We suppose that $\boldsymbol{\theta}_G$ holds $\log \pi(\boldsymbol{\theta}_G|\boldsymbol{\lambda}) = O(n)$. We can choose the optimal graph such that $p(G|\mathbf{X}_n)$ is maximum. A crucial problem for constructing a criterion based on the posterior probability of the graph is the computation of the high dimensional integration in Eq. 24.11. Heckerman and Geiger [31] used the conjugate priors for solving the integral and gave a closed-form solution. To compute this high dimensional integration, we use Laplace's approximation [14, 30, 75] for the integral

$$\int \prod_{i=1}^{n} f(\mathbf{x}_i|\boldsymbol{\theta}_G)\pi(\boldsymbol{\theta}_G|\boldsymbol{\lambda})d\boldsymbol{\theta}_G = \frac{(2\pi/n)^{r/2}}{|J_\lambda(\hat{\boldsymbol{\theta}}_G)|^{1/2}} \exp\{nl_\lambda(\hat{\boldsymbol{\theta}}_G|\mathbf{X}_n)\}\{1 + O_p(n^{-1})\},$$

where r is the dimension of $\boldsymbol{\theta}_G$,

$$l_\lambda(\boldsymbol{\theta}_G|\mathbf{X}_n) = \sum_{i=1}^{n} \log f(\mathbf{x}_i|\boldsymbol{\theta}_G)/n + \log \pi(\boldsymbol{\theta}_G|\boldsymbol{\lambda})/n,$$

$$J_\lambda(\boldsymbol{\theta}_G) = -\partial^2\{l_\lambda(\boldsymbol{\theta}_G|\mathbf{X}_n)\}/\partial\boldsymbol{\theta}_G\partial\boldsymbol{\theta}'_G$$

and $\hat{\boldsymbol{\theta}}_G$ is the mode of $l_\lambda(\boldsymbol{\theta}_G|\mathbf{X}_n)$. Then, by taking minus twice logarithm of $p(G)p(\mathbf{X}_n|G)$, we define the Bayesian network and nonparametric regression criterion, named BNRC, for selecting a graph

$$\text{BNRC}(G) = -2\log p(G) - r\log(2\pi/n) + \log|J_\lambda(\hat{\boldsymbol{\theta}}_G)| - 2nl_\lambda(\hat{\boldsymbol{\theta}}_G|\mathbf{X}_n). \tag{24.12}$$

The optimal graph is chosen such that the criterion BNRC in Eq. 24.12 is minimal. The merit of the use of the Laplace method is that it is not necessary to consider the use of the conjugate prior distribution. Hence the modeling in the larger classes of distributions of the model and prior is attained.

The decompositions of the prior distribution of $\boldsymbol{\theta}_G$, $\pi(\boldsymbol{\theta}_G|\boldsymbol{\lambda}) = \prod_{j=1}^{p} \pi_j (\boldsymbol{\theta}_j|\boldsymbol{\lambda}_j)$, and prior probability of the graph, $p(G) = \prod_{j=1}^{p} p(L_j)$, yield BNRC as a decomposable score:

$$\text{BNRC}(G) = \sum_{j=1}^{p} \text{BNRC}_j, \tag{24.13}$$

where L_j is the subgraph of G consisting of X_j, $Pa(X_j)$ and the edges between them. By removing constant terms that are independent of model selection, we obtain

$$\begin{aligned}\text{BNRC}_j &= 2q_j + (n - 2q_j - 1)\log(2\pi\hat{\sigma}_j^2) + (2q_j + 1 - M_{j\cdot})\log n \\ &+ \sum_{k:X_k\in Pa(X_j)} \Bigg\{ \log(|\Lambda_{jk}|/|K_{jk}|) - (M_{jk} - 2)\log\beta_{jk} \\ &+ \frac{n\beta_{jk}}{\hat{\sigma}_j^2}\hat{\boldsymbol{\gamma}}'_{jk}K_{jk}\hat{\boldsymbol{\gamma}}_{jk}\Bigg\}\end{aligned}$$

with $M_{j\cdot} = \sum_k M_{jk}$ and $\Lambda_{jk} = B'_{jk}B_{jk} + n\beta_{jk}K_{jk}$. Here β_{jk}'s are the parameters of BNRC_j and are set by minimizing BNRC_j. The Hessian matrix is approximated by

$$\log|J_\lambda(\boldsymbol{\theta}_G)| \approx \sum_{k:X_k\in Pa(X_j)} \log\left| -\frac{\partial^2 l_{\lambda_j}(\boldsymbol{\theta}_j|\mathbf{X}_n)}{\partial\boldsymbol{\gamma}_{jk}\partial\boldsymbol{\gamma}'_{jk}} \right| + \log\left| -\frac{\partial^2 l_{\lambda_j}(\boldsymbol{\theta}_j|\mathbf{X}_n)}{\partial(\sigma_j^2)^2} \right|.$$

The details of the derivation of BNRC are in Imoto et al. [39]. This decomposition property is important for the efficient computation of the score of BNRC when we change the structure of the graph; this yields efficient structure learning algorithms in the next section.

24.2.3.3 Efficient Learning Algorithms for Network Structure

In general, the problem of structure learning of Bayesian networks based on optimizing score function is known NP-hard [13]. Since gene networks usually contain several hundreds or more genes, for its learning, algorithms based on greedy heuristics are usually employed. Heuristic learning algorithms give us locally optimal structures that are not guaranteed as the global optima. In a greedy hill-climbing algorithm (HC), we test (1) adding one edge, (2) remove one edge and (3) reverse one edge direction and check the score. If the score is improved, we perform the best one and update the graph. HC repeats this procedure until the score converges. However, the time complexity of the above trials in one step is $O(p^2)$ and more than $O(p^3)$ in total, that is computationally hard if the number of genes is large. Therefore, like the sparse candidate algorithm [17], we restrict the number of candidate parents for each gene by m $(m \ll p)$ and apply the above trials for each gene and its candidates of parents; this yields $O(p)$ in one step of one gene. Note that the resulting structure based on this constrained greedy search depends on the order of genes to be learned. We usually test many permutations and take the best one from the learned networks.

For small Bayesian networks, an optimal search algorithm [56] can be used in time complexity $O(p2^p)$ and learn gene networks having 30 or less genes in the current computational capacity. Recently, a hybrid algorithm that learns a skeleton with an independency test approach and constrains on the directed acyclic graphs considered during the search-and-score phase, is shown to improve sensitively accuracy and speed [77]. According to the concept of hybrid approach, Perrier et al. [59] proposed an algorithm that can learn optimal Bayesian network when the undirected graph is given as the structural constraint. Perrier et al. [59] called the undirected graph as the super-structure; the skeleton of the learnt Bayesian network is a subgraph of the super-structure. The algorithm can learn optimal Bayesian network with 50 nodes when the average degree of the undirected graph is around two, i.e., sparse structural constraint.

For structure learning of dynamic Bayesian networks, we can ignore the acyclicity of the graph; the networks can allow cycles. Therefore, it is enough to find the best set of the parents for each gene. Since a simple enumeration requires exponential order of time complexity, we can use efficient algorithms like a branch and bound algorithm, which can reduce the computational time practically.

In learning dynamic Bayesian networks, another way for finding gene network structure is to use lasso [74]. The lasso uses L_1-type shrinkage in the loss function of the parameter estimation; it is equivalent to use a Laplace distribution as the prior distribution for the coefficients of linear regression and achieves parameter estimation and structure learning, simultaneously. As an extension, Kojima et al. [47] used additive nonparametric regression model for vector autoregressive models and estimated their parameters by group lasso. Furthermore, since many genes in microarray data are correlated each other and the lasso usually chooses the best one from the correlated genes; this yields unstable parameter estimates that cannot capture the group effects of genes. In such a case, elastic net [83] is advocated and can be used for gene network estimation [67].

Finally, we describe practical ideas for implementation of Bayesian network and nonparametric regression efficiently. Here we pick up three main tips that are employed in our implementation and effective to reduce the computational time.

1. In the BNRC score, the main calculation is to estimate coefficients of B-splines, γ_{jk}. These are calculated by the back fitting algorithm, which repeatedly calculates γ_{jk} until the mode $\hat{\sigma}_j^2$ converges. In the back fitting algorithm, the term $(B'_{jk}B_{jk}+n\beta_{jk}K_{jk})^{-1}B'_{jk}$ in Eq. 24.9 does not change during the network estimation. Therefore this can be calculated for every gene and stored in advance. We can avoid the computation of inverse matrices during the network estimation. The resultant matrix ($M_{jk} \times n$-size), however, requires relatively a large amount of memory to store. Therefore, if the amount of memory is insufficient, then $(B'_{jk}B_{jk} + n\beta_{jk}K_{jk})^{-1}$ ($M_{jk} \times M_{jk}$-size) can be stored instead of $(B'_{jk}B_{jk} + n\beta_{jk}K_{jk})^{-1}B'_{jk}$.
2. In the greedy hill-climbing algorithm, the local scores of the same genes with the same set of parents are repeatedly calculated. Therefore, the calculated score can be stored and refused many times during the network estimation. This significantly decreases the computational time.
3. Also in the greedy hill-climbing algorithm, it requires to check if the constructing graph is a DAG every time it adds or reverses an edge. To do this efficiently, the online version of the topological ordering algorithm can be used, such as the PK algorithm [57]. There exists a topological order that corresponds to a DAG. Thus if adding or reversing an edge does not affect the topological order of the current graph structure, then the edge does not affect the graph cyclicity and this can be done in a constant time.

24.2.4 Combining Prior Knowledge for Gene Networks with Microarray Data

A drawback in the gene network construction from microarray data is that while the gene network contains a large number of genes, the information contained in gene expression data is limited by the number of microarrays, their quality, the experimental design, noise, and measurement errors. Therefore, estimated gene networks contain some incorrect gene regulations, which cannot be evaluated from a biological viewpoint. In particular, it is difficult to determine the direction of gene regulation using gene expression data only. Hence, the use of biological knowledge, including protein-protein and protein-DNA interactions [7,10,24,34,43], sequences of the binding site of the genes controlled by transcription regulators [48,63,81], literature and so on, are considered to be a key for microarray data analysis. In this section, we provide a general framework for combining microarray data and biological knowledge aimed at estimating gene networks by using Bayesian network model. The key idea is to construct prior probability of the network represented by $p(G)$ in Eq. 24.10 by such kinds of prior knowledge for gene networks.

In order to combine various types of genomic data with microarray data for estimating gene networks, Imoto et al. [36] proposed a general framework that uses additional knowledge for gene networks as a prior probability of the graph in the context of Bayesian statistics. They considered prior biological knowledge as discrete information and construct a prior probability of the graph, denoted by $p(G)$ in the previous sections, based on the Gibbs distribution; the balance between the prior knowledge and microarray was tuned by the information criterion BNRC. According to this concept, Bernard and Hartemink [9] constructed $p(G)$ using the binding location data [48] that is a collection of p-values (continuous information). In this section, we construct $p(G)$ by using multi-source information including continuous and discrete prior information [41].

Let $\mathbf{Z}_k$ is the matrix representation of kth prior information, where (i, j)th element $z_{ij}^{(k)}$ represents the information of "gene $i \to$ gene j". For example,

1. If we use a prior network $G_{\text{prior}} = (\mathscr{X}, \mathscr{E}_{\text{prior}})$ for $\mathbf{Z}_k$, $z_{ij}^{(k)}$ takes 1 if $e(i, j) \in \mathscr{E}_{\text{prior}}$ or 0 if $e(i, j) \notin \mathscr{E}_{\text{prior}}$.
2. By using the gene knock-down data for $\mathbf{Z}_k$, $z_{ij}^{(k)}$ represents the value that indicates how gene j changes by knocking down gene i. We can use the absolute value of the log-ratio of gene j for gene i knock-down data as $z_{ij}^{(k)}$.
3. For a transcription factor (gene i), if ChIP-chip data are available, we can use minus log p-value of gene j as $z_{ij}^{(k)}$.

Using the adjacent matrix $E = (e_{ij})_{1 \le i,j \le p}$ of $G = (\mathscr{X}, \mathscr{E})$, where $e_{ij} = 1$ for $e(i, j) \in \mathscr{E}$ or 0 for otherwise, we assume the Bernoulli distribution on e_{ij} having probabilistic function

$$p(e_{ij}) = \pi_{ij}^{e_{ij}} (1 - \pi_{ij})^{1-e_{ij}},$$

where $\pi_{ij} = \Pr(e_{ij} = 1)$. For constructing π_{ij}, we use the logistic model

$$\pi_{ij} = 1/\{1 + \exp(-\eta_{ij})\}$$

with linear predictor

$$\eta_{ij} = \sum_{k=1}^{K} w_k (z_{ij}^{(k)} - c_k),$$

where w_k and c_k $(k = 1, \dots, K)$ are weight and baseline parameters, respectively. We then define a prior probability of the graph based on prior information $\mathbf{Z}_k$ $(k = 1, \dots, K)$ by

$$p(G) = \prod_i \prod_j p(e_{ij}).$$

This prior probability of the graph assumes that edges $e(i, j)$ $(i, j = 1, \dots, p)$ are independent of each other. In reality, there are several dependencies among e_{ij}'s such as $p(e_{ij} = 1) < p(e_{ij} = 1 | e_{ki} = 1)$ and so on. We may consider adding such information into $p(G)$ as an extension.

24.3 Computational Drug Target Discovery Using Microarray Data of HUVEC Treated with Fenofibrate

24.3.1 Data Sets

All data in this chapter were measured by CodeLink$^{\text{TM}}$ Human Uniset I 20K (20,469 probes).

24.3.1.1 Fenofibrate Time-Course Data

We measure the time-responses of human endothelial cell genes to 25 μM fenofibrate. The expression levels of 20,469 probes are measured at six time-points (0, 2, 4, 6, 8 and 18 h). Here time 0 means the start point of this observation and just before exposure to the fenofibrate. In addition, we measure this time-course data as the triplicate data in order to confirm the quality of experiments.

24.3.1.2 Gene Knock-Down Data by siRNA

For estimating gene networks, we newly created 400 gene knock-down data by using siRNA. We measure 20,469 probes for each knock-down microarray after 24 h of siRNA transfection. The knock-down genes are mainly transcription factors and signaling molecules. Let $\tilde{\mathbf{x}}_{D_i} = (\tilde{x}_{1|D_i}, \dots, \tilde{x}_{p|D_i})'$ be the raw intensity vector of ith knock-down microarray. For normalizing expression values of each microarray, we compute the median expression value vector $\mathbf{v} = (v_1, \dots, v_p)'$ as the control data, where $v_j = \text{median}_i(\tilde{x}_{j|D_i})$. We apply the loess normalization method to the MA transformed data, where $M_i = \log \tilde{x}_{i|D_j} - \log v_i$ and $A_i = (\log \tilde{x}_{i|D_j} + \log v_i)/2$ are plotted as $(A_i, M_i) \in R^2$, and the normalized intensity $x_{j|D_i}$ is obtained by applying the inverse transformation to the normalized $\log(\tilde{x}_{j|D_i}/v_j)$.

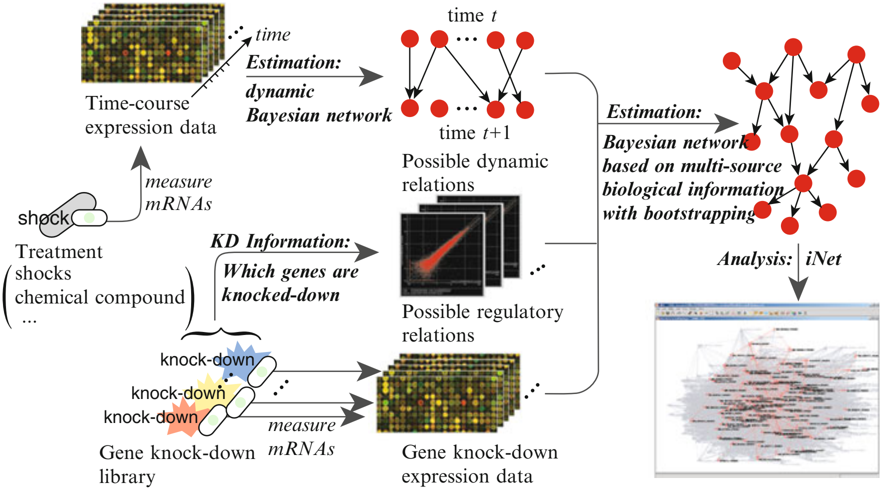


Fig. 24.1 Overview of the analysis for finding fenofibrate-related gene networks

24.3.2 Estimation of Gene Network Induced by Fenofibrate

24.3.2.1 Background and Overview

As a real application of gene network estimation techniques, computational drug target discovery enhanced with gene network inference [15, 38, 61, 73] has made tremendous impacts on pharmacogenomics. In this section, we show a proof-of-concept study of discovering druggable gene networks, which are most strongly affected by a chemical compound. For this purpose, we use two types of microarray data described previously: One is gene expression data obtained by measuring transcript abundance responses over time following treatment with the chemical compound. The other is gene knock-down expression data, where one gene is knocked-down for each microarray. Figure 24.1 is the conceptual view of our strategy in this section. First, we estimate dynamic relationships denoted by G_T between genes based on time-course data by using dynamic Bayesian networks [46]. Second, in gene knock-down expression data, since we know the information of knocked-down genes, possible regulatory relationships between knocked-down gene and its regulatees can be obtained. We denote this information by R. Finally, the gene network G_K is estimated by gene knock-down data denoted by $\mathbf{X}_K$ together with G_T and R by using Bayesian network based on multi-source biological information [39]. The key idea for estimating a gene network based on multi-source biological information is to use G_T and R as the Bayesian prior probability of G_K introduced in Sect. 24.2.4.

24.3.2.2 Selection of Genes Affected by Fenofibrate

For estimating fenofibrate-related gene networks from fenofibrate time-course data and 270 gene knock-down data (note that this analysis was done in Imoto et al. [41] and at that time we had this number of knock-down data as a part of our current data), we first define the set of genes that are possibly related to fenofibrate as follows: First, we extract the set of genes whose variance-corrected log-ratios, $|\log(x_{j|D_i}/v_j)/s_j|$, are greater than 1.5 from each time point, where $s_j = \mathrm{Var}[\log(x_{j|D_i}/v_j)|\log(x_{j|D_i} \cdot v_j)]$. We then find significant clusters of selected genes using GO Term Finder. Table 24.1 shows the significant clusters of genes at 18 h. The first column indicates how expression values are changed, i.e. "↗" and "↘" mean "overexpressed" and "suppressed", respectively. The GO annotations of clusters with "↘" are mainly related to cell cycle, the genes in these clusters are expressed ubiquitously and this is a common biological function. On the other hand, the GO annotations of clusters with "↗" are mainly related to lipid metabolism. In biology, it is reported that the fenofibrate acts around 12 h after exposure [21, 29]. Our first analysis for gene selection suggests that fenofibrate affects genes related to lipid metabolism and this is consistent with biological facts. We also focus on the genes from the 8 h time-point microarray. Unfortunately, no cluster with specific function could be found in the selected genes from the 8 h time-point microarray However, there also exist some genes related to lipid metabolism.

Table 24.1 Significant GO annotations of selected fenofibrate-related genes from the 18 h time-point microarray

		GO Function	p-value	Number of genes
↘	GO:0007049	Cell cycle	1.0E-08	35
↘	GO:0000278	Mitotic cell cycle	3.7E-07	19
↘	GO:0000279	M phase	5.0E-06	17
↗	GO:0006629	Lipid metabolism	1.3E-05	25
↘	GO:0007067	Mitosis	1.3E-05	15
↘	GO:0000087	M phase of mitotic cell cycle	1.6E-05	15
↘	GO:0000074	Regulation of cell cycle	2.7E-05	22
↗	GO:0044255	Cellular lipid metabolism	4.4E-05	21
↗	GO:0016126	Sterol biosynthesis	4.3E-04	6
↗	GO:0016125	Sterol metabolism	4.5E-04	8
↗	GO:0008203	Cholesterol metabolism	1.5E-03	7
↗	GO:0006695	Cholesterol biosynthesis	2.4E-03	5
↗	GO:0008202	Steroid metabolism	3.6E-03	10
↘	GO:0000375	RNA splicing, via transesterification reactions	4.1E-03	9
↘	GO:0000377	RNA splicing, via transesterification reactions with bulged adenosine as nucleophile	4.1E-03	9
↘	GO:0000398	Nuclear mRNA splicing, via spliceosome	4.1E-03	9
↗	GO:0006694	Steroid biosynthesis	6.0E-03	7
↘	GO:0016071	mRNA metabolism	6.3E-03	13

Therefore we use the genes from the 8 and 18 h time-point microarrays. Finally we add the 267 knock-down genes (three genes are not spotted on our chips) to the selected genes above, total 1,192 genes are defined as possible fenofibrate-related genes and used for the next network analysis.

24.3.2.3 Discovering Master-Regulator Genes in Fenofibrate Induced Gene Network

By converting the estimated dynamic network and knock-down gene information into the matrix representations of the first and second prior information $\mathbf{Z}_1$ and $\mathbf{Z}_2$, respectively, we estimate the gene network $\hat{G}_K$ based on $\mathbf{Z}_1, \mathbf{Z}_2$ and the knock-down data matrix $\mathbf{X}_K$. For extracting biological information from the estimated gene network, we first focus on lipid metabolism-related genes, because the clusters related this function are significantly changed at the 18 h time-point microarray. In the estimated gene network, there are 42 lipid metabolism-related genes and PPARα (*Homo sapiens* peroxisome proliferative activated receptor, alpha) is the only transcription factor among them. Therefore, we next focus on the node downstream of PPARα (Fig. 24.2). Among the candidate regulatees of PPARα, there are 21 lipid

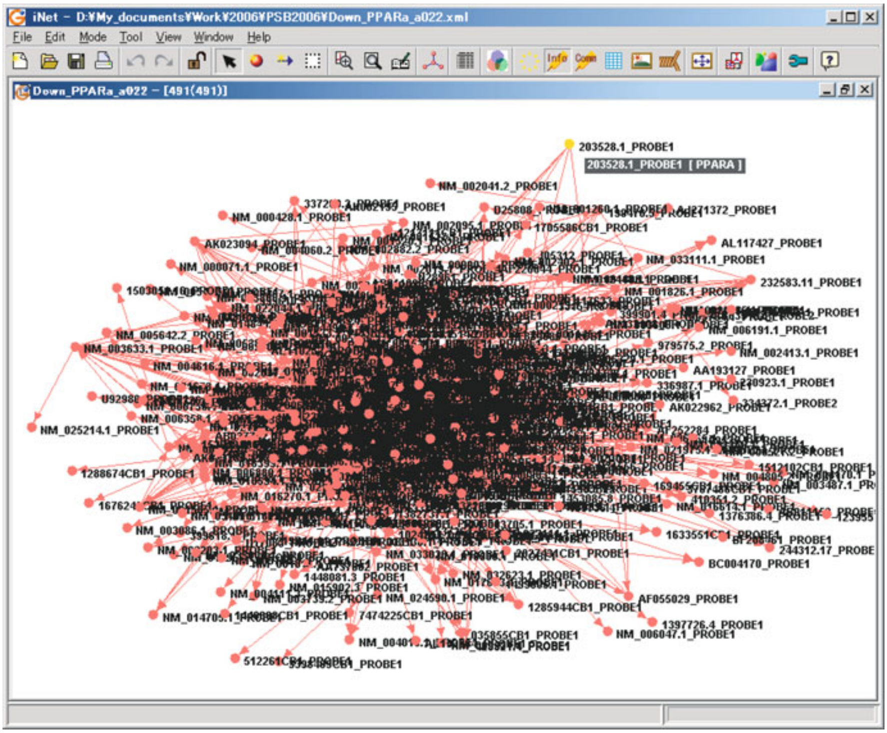


Fig. 24.2 Downstream of PPARα. There are 491 genes in four steps downstream. We consider these genes are candidate regulatees of PPARα

metabolism-related genes and 11 molecules previously identified experimentally to be related to PPARα. Actually, PPARα is known to be activated by fenofibrate.

One of the drug efficacies of fenofibrate is to reduce LDL cholesterol. LDLR and VLDLR are on the downstream of PPARα and mainly contribute to the transporting of cholesterol; they are candidate regulatees of PPARα. As for LDLR, it has been reported the relationship with PPARα [42]. We also could extract STAT5B and GLS that are children of PPARα and have been reported their regulation-relationships with PPARα [44, 69]. Therefore, it is not surprising that our network shows that many direct and indirect relationships involving known PPARα regulatees are triggered in endothelial cells by fenofibrate treatment. In the node upstream of PPARα, PPARα and RXRα, which form a heterodimer, share a parent. We could extract fenofibrate-related gene network and estimate that PPARα is the one of the key molecules of fenofibrate regulations without previous biological knowledge.

In the estimated fenofibrate induced network, there are 42 genes that are related to lipid metabolism. Among them, 17 genes have more children than PPARα. In these 17 genes, six genes are known as drug targets in pharmaceutical companies or having druggable motifs reported in Hopkins and Groom [33]. For example, the seventh hub gene is HMGCR, which is known as the target of pravastatin, another anti-hyperlipidaemia drug.

24.3.3 Discovery of Signaling Pathways Affecting Fenofibrate-Induced Gene Networks

24.3.3.1 Background and Overview

Drug-response pathways at a transcriptome level are successfully predicted by cutting-edge computational techniques. On the other hand, some drugs affect the pathways at protein level. For example, drugs affect secretion of secreted proteins (e.g., cytokines and growth factors) which are released from target cells. There is a possibility that these proteins have effects on target cells through drug-effected autocrine pathways. From the drug development viewpoint, these pathways could be useful for revealing drug mechanism of action, potentiation of drug effects and avoidance of side effects.

To validate the existence of such drug-affected autocrine pathways, we propose a novel computational method for finding signaling pathways that have the potential to regulate gene networks. The method combines gene networks estimated as drug-response pathways from mRNA expression data with proteome networks represented by protein-protein interactions to extract such pathways. First, we estimate a dynamic gene network from drug-response time-course microarray data by dynamic Bayesian network with nonparametric regression. For this, we propose the node-set separation method that enables us to find subnetworks significantly activated at observed time points, master-regulator genes and critical paths in the drug-response pathways. We then combine protein-protein interaction (PPI)

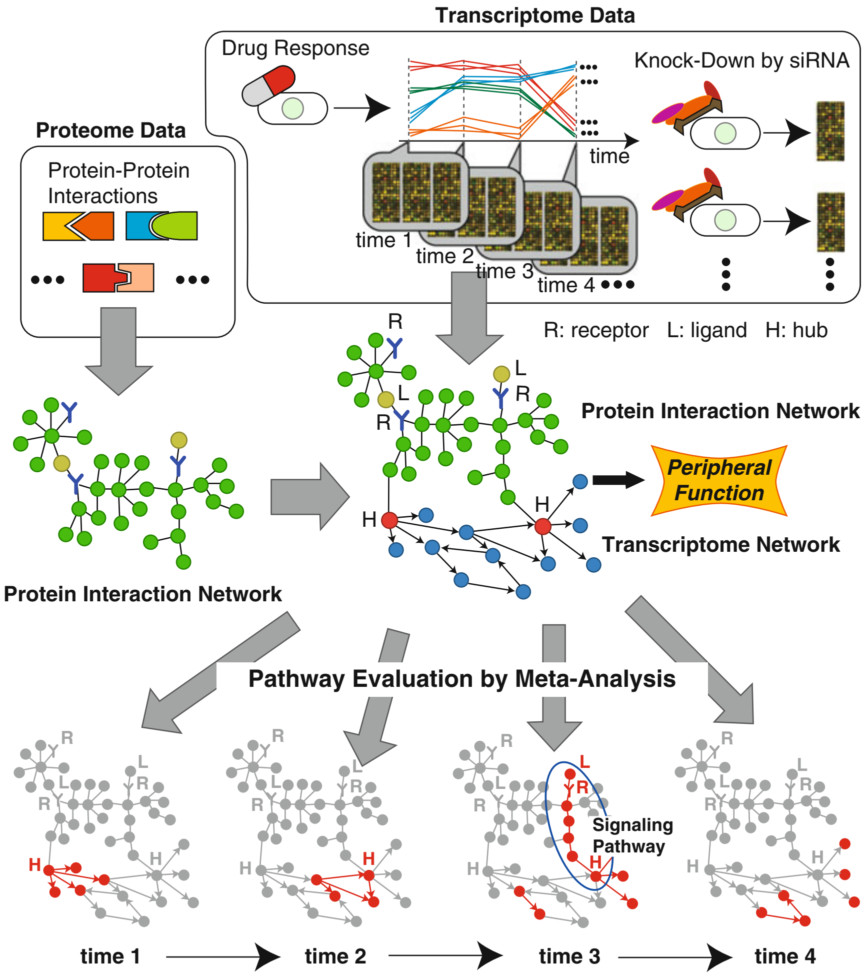


Fig. 24.3 Overview of a computational strategy for discovering signaling pathways affecting gene networks

network with the estimated dynamic gene network. The candidate signaling pathways that connect a ligand or a receptor to the key genes in the gene network are extracted and evaluated based on statistical hypothesis testing at each observed time. Based on the computed p-values, the candidate drug-affected autocrine pathways are selected by multiplicity corrected significance level.

Figure 24.3 represents the overview of the proposed method. Based on drug response time-course microarray data, we estimate a dynamic gene network by the dynamic Bayesian network model with nonparametric regression. However, ordinary dynamic Bayesian network can estimate a network from time-course data, while at each observed time-point, different sub-networks have high activity and transmit information of external signals to other sub-networks. Therefore, we need to extend dynamic Bayesian network to capture this feature. We introduce this extension in the next section.

24.3.3.2 Node-Set Separation Method for Dynamic Bayesian Network with Time-Dependent Structure Changes

The key idea of our dynamic Bayesian network estimation, called node-set separation method, is to define the active gene set for each time point. That is, a gene in an active gene set is determined as a differentially expressed gene comparing to the controls. Let $\mathscr{A}_t = \{g_i : \mathrm{pv}(g_i, t) \leq \theta_t\}$ be the active gene set at time t for $t = 1, \ldots, T$, where g_i represents the ith gene, $\mathrm{pv}(g_i, t)$ is the p-value of g_i at time t, and θ_t is the threshold for time t that could be determined by using false discovery rate for example. In our case, the p-value of each gene is computed by comparing triplet expression values of the gene at a time to control four replicate expression values, i.e., expression data of non-treated cells. We then define the *node set* $\mathscr{N}_t = \mathscr{A}_{t-1} \cup \mathscr{A}_t$ for $t = 1, \ldots, T$, where $\mathscr{A}_0$ is the empty set.

The definition of the node set has the basis on the Markov process of the dynamic Bayesian networks, i.e., the dynamic Bayesian network assumes the first order Markov process among time-course data. The gene network at time t, we denote G_t, is estimated for the node set $\mathscr{N}_t$ by the dynamic Bayesian network and nonparametric regression with whole 400 knock-down microarray gene expression data $\mathbf{X}_1, \ldots, \mathbf{X}_T$ [46]. Finally the dynamic gene network is obtained by $G = G_1 \cup \cdots \cup G_T$. The advantage of this estimation procedure, i.e., using node set $\mathscr{N}_t$ separately, by comparing with other algorithms that use $\mathscr{N} = \mathscr{A}_1 \cup \cdots \cup \mathscr{A}_T$ as the node set is not only finding dynamics of transcriptome networks, but also has a possibility to reduce false positive edges in the networks, because we can reduce the size of the gene set for each observed time efficiently; this can increase the accuracy of the structure learning.

First, we define master-regulator genes in each node set $\mathscr{N}_t$, based on the estimated G_t for $t = 1, \ldots, T$. The hub genes in $\mathscr{N}_t$ are defined as the top 5% genes; the genes in $\mathscr{N}_t$ are ranked according to the numbers of their direct child-genes in G_t. We denote the set of hub genes of $\mathscr{N}_t$ as $\mathscr{H}_t$. We also focus on the direct parents of the hub genes and represent the set of parent genes of the hub genes in $\mathscr{H}_t$ as $\mathscr{P}_t$. Since the hub genes and their direct parents could control the transcription levels of many genes in $\mathscr{N}_t$, we thus define the set of master-regulator genes at time t by $\mathscr{M}_t = \mathscr{H}_t \cup \mathscr{P}_t$.

Table 24.2 summarizes the numbers of nodes ($|\mathscr{N}_t|$), edges, hubs ($|\mathscr{H}_t|$), and hubs and their parents ($|\mathscr{M}_t|$) in the dynamic transcriptome network G_t. These hub and their parent genes were used as target nodes of the pathway extraction in the later step.

Table 24.2 Summary of the dynamic gene networks. $|\mathscr{H}_t|$ is the number of hub genes, $|\mathscr{M}_t|$ the number of hubs and their parents in G_t

G_t	nodes	edges	$\lvert\mathscr{H}_t\rvert$	$\lvert\mathscr{M}_t\rvert$
1 (2 h)	14	59	1	9
2 (2 h/4 h)	19	91	1	2
3 (4 h/6 h)	144	625	7	31
4 (6 h/8 h)	200	874	10	42
5 (8 h/18 h)	454	1 982	22	51

Table 24.3 The number of possible pathways s_{5k} and the final significant pathways with $\mathrm{pv}(s_{5k}, 5) < \xi_t$ with respect to the maximum distance l

l	all	final
1	13	3
2	651	3
3	27,373	43
4	1,194,215	150
5	51,078,582	806

24.3.3.3 PPI Paths for Candidate of Signaling Pathways

We then focus on the PPI network for exploring candidates of signaling pathways affecting master-regulator genes. On the PPI network, for $g_i \in \mathscr{M}_t$, we search receptors and ligands, denoted by r_j, that connect g_i by l or less edges, i.e., g_i connects with r_j by $l-2$ or less intermediate proteins. We denote the kth PPI path for the genes in $\mathscr{M}_t$ ending at $g_i \in \mathscr{M}_t$ as $s_{tk} = r_j - p_1 - p_2 - \cdots - g_i$, where p_1 and p_2 represent the intermediate proteins in the PPI network.

We checked the number of possible pathways to determine the appropriate l (maximum distance). Table 24.3 shows the number of all possible pathways from ligands or receptors to $\mathscr{M}_5$ (the hubs and their parents in 8 h/18 h transcriptome network G_5) evaluated by p-values of 18 h fenofibrate time course gene expression data ($\mathrm{pv}(s_{5k}, 5)$). According to this table, we decided to use $l = 4$ since it seems to be the most realistic and appropriate for the later analysis.

24.3.3.4 P-Values for PPI Paths by Meta-Analysis

Let $[p_i]$ represent the gene for the ith protein in the PPI network, i.e., if p_i is a protein translated from the i'th gene, we have $[p_i] = g_{i'}$. We also define $[r_j]$ in the same way. We assess the significance of s_{tk} using the p-values, $\mathrm{pv}([p'], t)$ for $p' \in s_{tk} \backslash \{g_i\}$, by statistical meta-analysis [25]. That is, we regard the p-value of each genes in s_{tk} as an evidence whether the PPI pathway s_{tk} is activated or not. We use the statistical meta-analysis method for integrating p-values of genes in s_{tk} into the p-value of s_{tk}.

The integrated p-value for s_{tk} is computed under the null hypothesis: all p-values $\mathrm{pv}([p'], t)$ are not significant, and the alternative hypothesis: at least one or more p-values $\mathrm{pv}([p'], t)$ are significant. That is, if the null hypothesis is not rejected,

s_{tk} seems to be not functional; otherwise if we observe the small p-value, s_{tk} is activated and is functional. For the meta-analysis, we use Fisher's inversion method to integrate p-values. We remove the p-value of g_i for the meta-analysis, because g_i was selected as a significant genes in $\mathscr{N}_t$. Therefore, it is obvious that s_{tk} is decided as significant if g_i is included in the meta-analysis calculation, and is meaningless.

Since the node set $\mathscr{N}_t$ is constructed by the active gene sets of time t and $t-1$, there are two ways to assess the significance of s_{tk} by using either p-value at time t or $t-1$. We test both cases and assess the significance of each PPI path. We determine s_{tk} is significant if and only if either $\mathrm{pv}(s_{tk}, t) < \xi_t$ or $\mathrm{pv}(s_{tk}, t-1) < \xi_{t-1}$ holds, where $\mathrm{pv}(s_{tk}, t)$ is the integrated p-value of the PPI path s_{tk} with p-values at time t and ξ_t is the threshold determined by considering multiplicity of the testings. In the real data analysis, we use 1% significant level with the Bonferroni correction. Obviously, other methods for controlling multiplicity of testing, such as family-wise error rate, false discovery rate and so on, can be used for reducing false negatives. The reason why we choose the Bonferroni method is that since we use the results of statistical tests for mRNA expression data for finding the significance of protein levels, some changes of protein levels are not measured normally. Therefore, we choose the most strict correction method to achieve a conservative method.

In order to confirm that the method can capture known pathways related to fenofibrate, we focused on PPI paths related to PPARα, since PPARα is a target of fenofibrate. In the dynamic transcriptome network analysis, PPARα is included in the node sets $\mathscr{N}_4$ and $\mathscr{N}_5$, i.e., PPARα was over-expressed at 8 and 18 h. In both times, PPARα was selected as a hub gene. In G_4 PPARα has 21 children and 31 in G_5. Since we would like to investigate drug-affected autocrine pathways, we first limited the candidate PPI paths by autocrine ligand pathways (ALPs) that connect ligands included in earlier time gene networks, i.e., active ligands in earlier times, to hub genes and their parent genes.

By the Bonferroni correction with 1% significance level, only 23 pathways from ligands in G_3 or G_4 to $\mathscr{M}_5$ evaluated by 8 h expression data remained as significant ALPs (Table 24.4). Among them, we found that the pathway including PPARα as a hub gene of the gene network has high statistical significance (the fourth highest significance). This ALP is VEGF–NRP1–GIPC1–PRKCA–PPARα. PRKCA, protein kinase C alpha, is located on the upstream of PPARα. PRKCA is one of the

Table 24.4 The numbers of all the ALPs from ligands in $G_{t'}$ to $\mathscr{M}_t$

$G_{t'}$ ligands	$\mathscr{M}_t$	total	eval $t-1$	eval t
2 (2 h/4 h)	3 (4 h/6 h)	437	35	126
2 (2 h/4 h)	4 (6 h/8 h)	894	160	27
3 (4 h/6 h)	4 (6 h/8 h)	1,448	177	28
2 (2 h/4 h)	5 (8 h/18 h)	533	30	27
3 (4 h/6 h)	5 (8 h/18 h)	873	23	23
4 (6 h/8 h)	5 (8 h/18 h)	873	23	23

Column "total" represents the number of all possible pathways ($l = 4$). Columns "eval $t-1$" and "eval t" are the numbers of ALPs that are statistically significant if evaluated by $\mathrm{pv}(s_{tk}, t-1)$ and $\mathrm{pv}(s_{tk}, t)$, respectively.

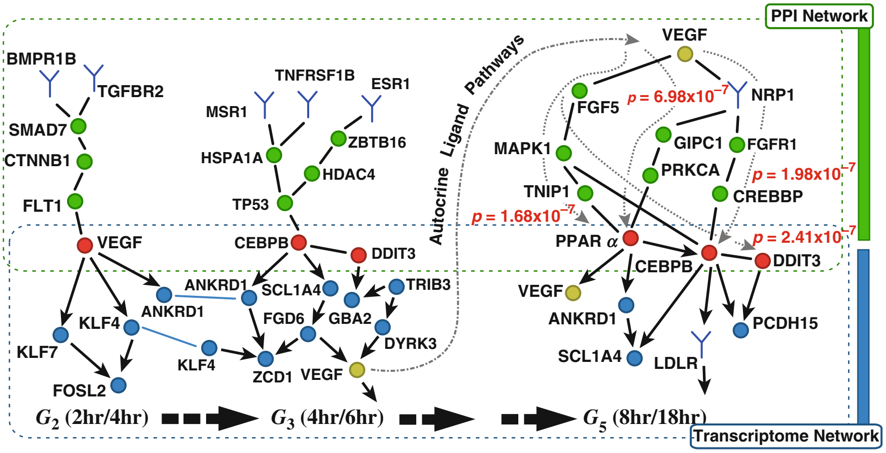


Fig. 24.4 The top four autocrine pathways and their p-values, which are connected to G_5 (8 h/18 h gene network) hub and their parent genes. Some parts of gene networks and the significant pathways at the previous times are also presented to illustrate dynamical changes extracted by the proposed method

members of serine- and threonine-specific protein kinases and is related to phospholyration of many genes including PPARα. Protein kinase C inhibitor inactivates the phosphorylation of PPARα and induces the trans-repression activity of PPARα in hepatocytes. Our method was able to extract this known relationship, which is related to PPARα's trans-repression, with high statistical significance. VEGF, vascular endothelial growth factor A, is also included in this pathway. VEGF is a member of the PDGF/VEGF growth factor family and is the predominant regulator of angiogenesis. It has been reported that fenofibrate induces VEGF mRNA and prevents cell from apoptotic cell death in human retinal endothelial cells (HRECs). VEGF is also significantly up regulated in our microarray experiment. From this, our result suggests that the trans-repression property of fenofibrate might be caused by PRKCA mediated thorough VEGF signaling (Fig. 24.4).

24.3.4 Novel Drug Target Discovery Based on Gene Network Information

24.3.4.1 Background and Objective

PPARα is a ligand-activated transcription factor belonging to the family of nuclear receptors. PPARα is highly expressed in liver, skeletal muscle, kidney, and heart and regulates the transcription of genes involved in energy metabolism [20, 49]. Over the past decade, PPARα has been investigated as a therapeutic target and drugs targeting PPARα have been developed. Fenofibrate is one of the synthetic ligands of PPARα and has been widely used for the treatment of hyperlipidaemia, type 2 diabetes and cardiovascular diseases due to its the lipid-lowering effects [71].

A reported molecular mechanism of the lipid-lowering effect is “trans-activation”, that is, PPARα activated by fenofibrate forms a PPAR-RXR heterodimer complex, which binds to PPREs in the promoter regions of genes involved in beta-oxidation and lipoprotein/cholesterol transport [20]. In addition, fenofibrate also has anti-inflammatory and anti-atherogenic functions, which are thought to be based on “trans-repression” mechanisms in endothelial cells, smooth muscle cells and other vascular cells [82].

While the lipid-lowering molecular mechanisms in the liver are well known, the anti-inflammatory mechanisms in vascular cells have not been fully investigated. In addition, there are some PPARα-independent drug effects in human endothelial cells. For example, fenofibrate has been shown to regulate the survival of cultured human retinal endothelial cells in PPARα-independent manner, since pretreatment with the PPARα antagonist, MK 886, did not alter this effect and since another selective agonist for PPARα, WY-14643, had no significant effect on cell survival [45]. Moreover, in human umbilical vein endothelial cells (HUVECs), fenofibrate has been shown to increase AMPK phosphorylation, but neither bezafibrate nor WY-14643 had the same effect [51]. Therefore, we speculate that fenofibrate has PPARα-independent actions in human endothelial cells.

The objectives of this study are (1) to identify transcripts in HUVECs regulated by fenofibrate in a PPARα-dependent and a PPARα-independent manner: (2) to construct dynamic Bayesian gene networks to reveal PPARα-independent mechanisms of action of fenofibrate, and the master regulators of PPARα-independent transcripts, based on computational data analysis techniques.

24.3.4.2 Gene Selection

We first compared fenofibrate-treated cells in data set A in Fig. 24.5 to untreated control cells in data set N (comparison 1) and second compared PPARα siRNA-treated cells in data set C to control siRNA-treated cells in data set B (comparison 2). In the comparison 1, we identified fenofibrate-regulated transcripts based on the Significant Analysis of Microarray (SAM) statistical test, which takes multiple testing into account by estimating a false discovery rate, “q-value” [78]. If a transcript was up or down-regulated 1.7-fold or more in data set A rather than in data set N, and had a SAM q-value that was less than or equal to 0.05, we regarded this transcript as a fenofibrate-regulated transcript. The number of regulated transcripts gradually increased with time of fenofibrate exposure and fenofibrate’s effects were first clearly observed after 6 h of treatment which is consistent with previous report [21]. Interestingly, the transcript encoding PPARα was not detected to be up-regulated until after 8 h of treatment. Then, we used the MetaGP [25] tool to biologically interpret microarray data at each time point. MetaGP evaluated the significance of Gene Ontology terms based on the p-value at each time point. This analysis suggested that inflammatory related genes annotated by inflammatory response (GO:0006954), angiogenesis (GO:0001525) and cell adhesion (GO:0007155) were highly significant at middle or late time points (6, 8 and 18 h).

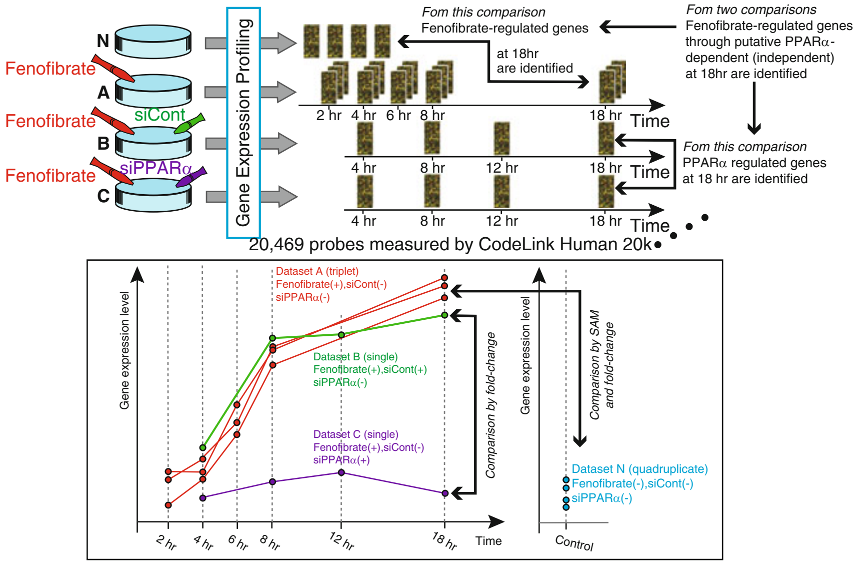


Fig. 24.5 Schematic view of this study. N: non-treated HUVECs ($n = 4$), A: HUVECs treated with 25 umol fenofibrate ($n = 3$), B: HUVECs adding control siRNA prior to the treatment of fenofibrate ($n = 1$), C: HUVECs adding PPARα siRNA prior to the treatment with fenofibrate ($n = 1$)

In the comparison 2, if a transcript was up- or down-regulated 1.7-fold or more in data set C rather than in data set B, we regarded this transcript as a PPARα-regulated transcript. From two comparisons, if a transcript was oppositely regulated by fenofibrate and PPARα siRNA, we regarded this transcript as a fenofibrate-regulated transcript through PPARα dependent mechanisms. The remaining fenofibrate-regulated transcripts that did not meet the criterion in the comparison 2 were considered PPARα-independently regulated transcripts. Contrary to our expectations, data for most fenofibrate-regulated transcripts were consistent with PPARα-independent fenofibrate mechanisms of action, in spite of fact that PPARα is a target of fenofibrate (Table 24.5). It is interesting to speculate about the possible explanations for this observation. Fenofibrate and other PPARα ligands appear to predominantly modulate the expression of genes that are traditionally thought to be activated by inflammatory pathways. These genes may not been significantly regulated by fenofibrate responses in HUVECs without a prior inflammatory stimulus, as was the case in our study. This could lead to our observation of mainly PPARα-independent regulation of transcript abundance by fenofibrate. Another possible explanation is that fenofibrate might mainly act through PPARα-independent mechanisms in endothelial cells. Unlike the mechanism of lipid-lowering actions based on the binding PPREs of promoter regions of PPARα targeting genes, the anti-inflammatory mechanisms may be very complicated.

Table 24.5 The number of transcripts oppositely regulated by fenofibrate and PPARα siRNA-mediated knock-down

Time points	4 h	8 h	18 h
Up regulation in fenofibrate & Down regulation in siPPARα	0	6	98
Down regulation in fenofibrate & Up regulation in siPPARα	0	7	66

24.3.4.3 Gene Network-Based Drug Target Discovery

We use dynamic Bayesian network and nonparametric regression method [46] with the node-set separation method [72] in order to find dynamical changes of activities of gene networks by dosing with fenofibrate. From this combination, we can find which networks are activated over our observed time course. In this study, we estimated five gene networks based on node-set separation method [72]. Each node-set consists of regulated transcripts at (1) 2 h, (2) 2 or 4 h, (3) 4 or 6 h, (4) 6 or 8 h, and (5) 8 and 18 h, respectively.

Here, we mainly focused on the gene network with transcript sets which are significantly regulated at 8 or 18 h, because PPARα is regulated at the same time points. We initially evaluated our gene network predictions by reference to our siRNA experiments in which PPARα was knocked down. We were reassured that 14 out of 28 PPARα child transcriptomes in the gene network show PPARα-dependent manner in our siRNA experiments (hypergeometric test p-value < 0.01). This result suggests that our gene network methods can capture at least a subset of PPARα regulated genes.

We next focused on the hub transcripts in our network. We defined hub transcripts as those transcripts with the top 5% of numbers of direct children in the network. Reassuringly, PPARα, a direct target of fenofibrate, is listed as a hub transcript. This seems reasonable because a drug targeted molecule is likely to be important for downstream gene-gene regulation relationships.

We also identified transcripts whose children in transcript network were significant enrichment of PPARα-independently regulated transcripts (Table 24.6). There are five transcripts including one EST or two hypothetical proteins which meet significant p-value < 0.01. Among them, solute carrier family 1 member 4 (SLC1A4), shows the highest significance (p-value < 1.2E-03). This gene is a glutamate/neutral amino acid transporter and mediates the efflux of L-serine from glial cells and its uptake by neurons [80], but its relevance to human endothelial cells has not been reported previously. Growth differentiation factor 15/macrophage inhibitory cytokines 1 (GDF15/MIC-1), is very interesting. This gene is a transforming growth factor beta family related protein that exerts multiple effects on cell fate such as on cell growth, differentiation, and inflammatory and apoptotic pathways [1] and is regulated by several anti-tumor agents. GDF15 inhibits endothelial cell migration and decreases matrix metallopeptidase 2 (MMP2) activity produced by the HUVECs in a concentration-dependent manner [16]. These effects are very similar to fenofibrate's effects. GDF15 is also listed as a hub transcript, therefore we

Table 24.6 Gene list having PPARα-dependently regulated transcripts as its child in gene network with statistical significance (p-value < 0.01)

Gene name	Gene description	A	B	p-value
SLC1A4	Solute carrier family 1 member 4	34	32	1.2E-03
LOC201895	Chromosome 4 open reading frame 34	75	64	3.4E-03
AK023999	EST	24	23	3.5E-03
LOC148189	Hypothetical protein LOC148189	85	37	6.2E-04
FST	Follistatin	23	22	4.8E-03
GDF15	Growth differentiation factor 15	34	31	5.5E-03

A: The number of child transcripts in gene network.
B: The number of PPARα-independent regulated transcripts in "A".

focused on the 34 downstream children of GDF15 and 11 out of these 34 transcripts are related to apoptosis and cell death.

24.4 Discussion

In this chapter, we show statistical inference of gene networks from microarray gene expression data. Bayesian networks extended for analyzing continuous data with nonlinear complex structure were introduced. For the statistical modelings in this chapter, the use of nonparametric regression with basis function expansion with B-splines is essential. An information criterion for its structure learning was derived from a Bayesian point of view. We computed the posterior probability of the graph by using Laplace's approximation and showed that the derived information criterion can be considered as an extension of Schwarz' BIC. For applying aforementioned statistical models and the information criterion, BNRC, for estimating gene networks, we need to establish a systematic procedure for determining structure of gene networks, because the structure learning of Bayesian networks is an NP-hard problem; a naive enumeration requires super-exponential time-complexity with respect to the number of genes. For this problem, we showed several computational algorithms including greedy heuristics and optimal search.

By using our gene network estimation technology, we showed three studies [6, 41, 72] in computational drug target discovery. All three studies use microarray data of human endothelial cells. We focused on the mode-of-action of an anti-hyperlipidaemia drug, fenofibrate, in HUVEC.

- In the first study, fenofibrate response time-course microarray data and single gene knock-down microarray data were combined and we focused on the hub genes in the estimated gene networks; we found that PPARα, a target of fenofibrate, is a hub gene and other known drug targets have also high connectivity. From the first analysis, we found that like protein-protein interaction networks hub genes in the estimated gene networks should be important for defining the molecular function of the estimated networks.

- In the second study, we tried to extract signaling pathways that strongly affect the action of fenofibrate in transcriptome. We extended the dynamic Bayesian network so that it can allow structure changes in time by the node-set separation method and we applied it to fenofibrate response time-course microarray data. For finding such signaling pathways, we combined protein interaction networks constructed by protein-protein interaction data and extracted significant signaling pathways from it. Based on this framework, we tested a biological hypothesis: there exist autocrine signaling pathways that are dynamically regulated by drug response transcriptome networks and control them simultaneously. In this result, from over one million possible protein-protein interaction pathways, we extracted significant 23 autocrine-like pathways with the Bonferroni correction, including VEGF−NRP1−GIPC1−PRKCA−PPARα, that is one of the most significant ones and contains PPARα.
- In the third study, we further evaluated the gene networks estimated by fenofibrate response time-course microarray data by using PPARα knock-down microarray data. We found that unexpectedly many of fenofibrate-regulated genes are not on the downstream of PPARα. By considering the estimated network topology around these genes, we found that GDF15 may be an important regulator of PPARα-independent fenofibrate action in HUVECs; that can be considered as a novel drug target.

For further analysis of molecular networks, it is important to combine statistical analysis described in this chapter with simulation model for molecular networks like Cell Illustrator [12, 53]. Since gene networks estimated by microarray gene expression data and statistical graphical models like Bayesian networks are a kind of approximate models of molecular networks, development of a more sophisticated model that can simulate the responses of molecular networks against various external stimuli or evaluate the variations of networks. Since simulation models of molecular networks are usually created by gathering knowledge provided from the literature, parameters such as speed of biological process, initial values and so on are sometimes tuned manually. Furthermore, the network structure constructed by the literature is often incomplete. For these problems, we proposed to use a statistical method called data assimilation that combines simulation model and observed data in order to automatic parameter estimation and network structure determination [54]. We consider that a strategic method that uses statistical network estimation method to determine key transcripts in the network, learns simulation model including genes and proteins from the literature and identifies key transcripts by data assimilation; this yields more efficient strategy to discover novel drug target pathways.

Acknowledgements The authors wish to thank Satoru Kuhara, Kousuke Tashiro, Masao Nagasaki, Yukiko Nakanishi, Atsushi Doi, Yuki Tomiyasu, Kaori Yasuda, Cristin Print, D. Stephen Charnock-Jones, Sally Humphreys, Ben Dunmore, Deborah Sanders, Christopher J. Savoie for their continuous efforts on our HUVEC study. Computation time was provided by the Super Computer System, Human Genome Center, Institute of Medical Science, University of Tokyo.

References

1. Ago, T., & Sadoshima J. (2006). GDF15, a cardioprotective TGF-beta superfamily protein. *Circulation Research*, *98*, 294–297.
2. Akaike, H. (1973). Information theory and an extension of the maximum likelihood principle. In B. N. Petrov & F. Csaki (Eds.), *Proc. 2nd International Symposium on Information Theory* (pp. 267–281). Budapest: Akademiai Kiadó.
3. Akutsu, T., Miyano, S., & Kuhara, S. (1999). Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Pacific Symposium on Biocomputing*, *4*, 17–28.
4. Akutsu, T., Miyano, S., & Kuhara, S. (2000). Algorithms for identifying Boolean networks and related biological networks based on matrix multiplication and fingerprint function. *Journal of Computational Biology*, *7*, 331–344.
5. Akutsu, T., Kuhara, S., Maruyama, O., & Miyano, S. (2003) Identification of genetic networks by strategic gene disruptions and gene overexpressions under a boolean model. *Theoretical Computer Science*, *298*, 235–251.
6. Araki, H., Tamada, Y., Imoto, S., Dunmore, B., Sanders, D., Humphrey, S., Nagasaki, M., Doi, A., Nakanishi, Y., Yasuda, K., Tomiyasu, Y., Tashiro, K., Print, C., Charnock-Jones, D. S., Kuhara, S., & Miyano, S. (2009). Analysis of PPAR alpha-dependent and PPAR alpha-independent transcript regulation following fenofibrate treatment of human endothelial cells. *Angiogenesis*, *12*(3), 221–229.
7. Bader, G. D., Donaldson, I., Wolting, C., Ouellette, B. F. F., Pawson, T., & Hogue, C. W. V. (2001). BIND-The biomolecular interaction network database. *Nucleic Acids Research*, *29*, 242–245.
8. Bannai, H., Inenaga, S., Shinohara, A., Takeda, M., & Miyano, S. (2002). A string pattern regression algorithm and its application to pattern discovery in long introns. *Genome Informatics*, *13*, 3–11.
9. Bernard, A., & Hartemink, A. (2005). Informative structure priors: Joint learning of dynamic regulatory networks from multiple types of data. *Pacific Symposium on Biocomputing*, *10*, 459–470.
10. BIND. http://www.blueprint.org/
11. Bussemaker, H. J., Li, H., & Siggia, E. D. (2001). Regulatory element detection using correlation with expression. *Nature Genetics*, *27*, 167–171.
12. Cell Illustrator: http://www.cellillustrator.com/
13. Chickering, D. (1996). Learning Bayesian networks is NP-complete. In *Learning from data: Artificial intelligence and statistics V* (pp. 121–130). Springer-Verlag.
14. Davison, A. C. (1986). Approximate predictive likelihood. *Biometrika*, *73*, 323–332.
15. Di Bernardo, D., Thompson, M. J., Gardner, T. S., Chobot, S. E., Eastwood, E. L., Wojtovich, A. P., Elliott, S. J., Schaus, S. E., & Collins, J. J. (2005). Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. *Nature Biotechnology*, *23*(3) 377–383.
16. Ferrari, N., Pfeffer, U., Dell'Eva, R., Ambrosini, C., Noonan, D. M., & Albini, A. (2005). The transforming growth factor-beta family members bone morphogenetic protein-2 and macrophage inhibitory cytokine-1 as mediators of the antiangiogenic activity of N-(4-hydroxyphenyl) retinamide. *Clinical Cancer Research*, *11*, 4610–4619.
17. Friedman, N., Nachman, I., & Pe'er, D. (1999). Learning Bayesian network structure from massive datasets: The "sparse candidate" algorithm. In *Fifteenth conference on uncertainty in artificial intelligence, UAI-99*.
18. Friedman, N., Linial, M., Nachman, I., & Pe'er, D. (2000). Using Bayesian network to analyze expression data. *Journal of Computational Biology*, *7*, 601–620.
19. Fujita, A., Sato, J. R., Garay-Malpartida, H. M., Yamaguchi, R., Miyano, S., Sogayar, M. C., & Ferreira, C. E. (2007). Modeling gene expression regulatory networks with the sparse vector autoregressive model, *BMC Systems Biology*, *1*, 39.

20. Gervois, P., Fruchart, J. C., & Staels, B. (2007). Drug insight: Mechanisms of action and therapeutic applications for agonists of peroxisome proliferator-activated receptors. *Nature Clinical Practice Endocrinology & Metabolism*, *3*, 145–156.
21. Goya, K., Sumitani, S., Xu, X., Kitamura, T., Yamamoto, H., Kurebayashi, S., Saito, H., Kouhara, H., Kasayama, S., & Kawase, I. (2004). Peroxisome proliferator-activated receptor α agonists increase nitric oxide synthase expression in vascular endothelial cells. *Arteriosclerosis, Thrombosis, and Vascular Biology*, *24*, 658–663.
22. Granger C. W. J. (1969). Investigating causal relationships by econometric models and cross-spectral methods. *Econometrica*, *37*, 424–438.
23. Green, P. J., & Silverman, B. W. (1994). *Nonparametric regression and generalized linear models*. Chapman & Hall/CRC Monographs on Statistics & Applied Probability.
24. GRID. http://biodata.mshri.on.ca/grid/servlet/Index
25. Gupta, P. K., Yoshida, R., Imoto, S., Yamaguchi, R., & Miyano, S. (2007). Statistical absolute evaluation of gene ontology terms with gene expression data. In *Proceedings of the 3rd international symposium on bioinformatics research and applications. Lecture note in bioinformatics* (Vol. 4463, pp. 146–157). Atlanta, Springer-Verlag.
26. Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., & Young, R. A. (2001). Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *Pacific Symposium on Biocomputing*, *6*, 422–433.
27. Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., & Young, R. A. (2002). Combining location and expression data for principled discovery of genetic regulatory network models. *Pacific Symposium on Biocomputing*, *7*, 437–449.
28. Hastie, T., & Tibshirani, R. (1990). *Generalized additive models*. Chapman & Hall.
29. Hayashida, K., Kume, N., Minami, M., Inui-Hayashida, A., Mukai, E., Toyohara, M., & Kita, T. (2004). Peroxisome proliferator-activated receptor alpha ligands activate transcription of lectin-like oxidized low density lipoprotein receptor-1 gene through GC box motif. *Biochemical and Biophysical Research Communications*, *323*, 1116–1123.
30. Heckerman, D. (1998). A tutorial on learning with Bayesian networks. In M. I. Jordan (Ed.). *Learning in Graphical Models*, Kluwer Academic Publisher.
31. Heckerman, D., & Geiger, D. (1995). Learning Bayesian networks: A unification for discrete and Gaussian domains. In *Proceedings of the eleventh conference on uncertainty in artificial intelligence* (pp. 274–284).
32. Hirose, O., Yoshida, R., Imoto, S., Yamaguchi, R., Higuchi, T., Charnock-Jones, D. S., Print, C., & Miyano, S. (2008). Statistical inference of transcriptional module-based gene networks from time course gene expression profiles by using state space models. *Bioinformatics*, *24*, 932–942.
33. Hopkins, A. L., & Groom, C. R. (2002). The druggable genome. *Nature Reviews Drug Discovery*, *1*, 727–730.
34. Ideker, T., Ozier, O., Schwikowski, B., & Siegel, A. F. (2002). Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, *18*(ISMB 2002), S233–S240.
35. Imoto, S., Goto, T., & Miyano, S. (2002). Estimation of genetic networks and functional structures between genes by using Bayesian networks and nonparametric regression. *Pacific Symposium on Biocomputing*, *7*, 175–186.
36. Imoto, S., Higuchi, T., Goto, T., Tashiro, K., Kuhara, S., & Miyano, S. (2003). Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. In *Proceedings of the IEEE 2nd computational systems bioinformatics (CSB2003)* (pp. 104–113).
37. Imoto, S., Kim, S., Goto, T., Aburatani, S., Tashiro, K., Kuhara, S., & Miyano, S. (2003). Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. *Journal of Bioinformatics and Computational Biology*, *1*, 231–252.
38. Imoto, S., Savoie, C. J., Aburatani, S., Kim, S., Tashiro, K., Kuhara, S., & Miyano, S. (2003). Use of gene networks for identifying and validating drug targets. *Journal of Bioinformatics and Computational Biology*, *1*(3), 459–474.

39. Imoto, S., Higuchi, T., Goto, T., Tashiro, K., Kuhara, S., & Miyano, S. (2004). Combining microarrays and biological knowledge for estimating gene networks via Bayesian networks. *Journal of Bioinformatics and Computational Biology*, *2*(1), 77–98.
40. Imoto, S., Higuchi, T., Goto, T., & Miyano, S. (2006). Error tolerant model for incorporating biological knowledge with expression data in estimating gene networks. *Statistical Methodology*, *3*(1), 1–16.
41. Imoto, S., Tamada, Y., Araki, H., Yasuda, K., Print, C. G., Charnock-Jones, D. S., Sanders, D., Savoie, C. J., Tashiro, K., Kuhara, S., & Miyano, S. (2006). Computational strategy for discovering druggable gene networks from genome-wide RNA expression profiles. *Pacific Symposium on Biocomputing*, *11*, 559–571.
42. Islam, K. K., Knight, B. L., Frayn, K. N., Patel, D. D., & Gibbons, G. F. (2005). Deficiency of PPARalpha disturbs the response of lipogenic flux and of lipogenic and cholesterogenic gene expression to dietary cholesterol in mouse white adipose tissue. *Biochimica et Biophysica Acta*, *1734*, 259–268.
43. Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., & Sakaki, Y. (2001). A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the Nationl Academy of Sciences of the United States of America*, *97*, 4569–4574.
44. Kersten, S., Mandard, S., Escher, P., Gonzalez, F. J., Tafuri, S., Desvergne, B., & Wahli, W. (2001). The peroxisome proliferator-activated receptor alpha regulates amino acid metabolism. *FASEB Journal*, *15*, 1971–1978.
45. Kim, J., Ahn, J. H., Kim, J. H., Yu, Y. S., Kim, H. S., Ha, J., Shinn, S. H., & Oh, Y. S. (2007). Fenofibrate regulates retinal endothelial cell survival through the AMPK signal transduction pathway. *Experimental Eye Research*, *84*, 886–893.
46. Kim, S., Imoto, S., & Miyano, S. (2004). Dynamic Bayesian network and nonparametric regression for nonlinear modeling of gene networks from time series gene expression data. *Biosystems*, *75*(1–3), 57–65.
47. Kojima, K., Fujita, A., Shimamura, T., Imoto, S., & Miyano, S. (2008). Estimation of nonlinear gene regulatory networks via L_1 regularized NVAR from time series gene expression data, *Genome Informatics*, *20*, 37–51.
48. Lee, T. I., Rinaldi, N. J., Robert, F., Odom, D. T., Bar-Joseph, Z., Gerber, G. K., Hannett, N. M., Harbison, C. T., Thompson, C. M., Simon, I., Zeitlinger, J., Jennings, E. G., Murray, H. L., Gordon, D. B., Ren, B., Wyrick, J. J., Tagne, J. -B., Volkert, T. L., Fraenkel, E., Gifford, D. K., & Young, R. A. (2002). Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, *298*, 799–804.
49. Lefebvre, P., Chinetti, G., Fruchart, J. C., & Staels, B. (2006). Sorting out the roles of PPAR in energy metabolism and vascular homeostasis. *Journal of Clinical Investigation*, *116*, 571–580.
50. Masys, D. R. (2001). Linking microarray data to the literature. *Nature Genetics*, *28*, 9–10.
51. Murakami. H., Murakami. R., Kambe. F., Cao. X., Takahashi. R., Asai. T., Hirai, T., Numaguchi, Y., Okumura, K., Seo, H., & Murohara, T. (2006). Fenofibrate activates AMPK and increases eNOS phosphorylation in HUVEC. *Biochemical and Biophysical Research Communications*, *341*, 973–978.
52. Murphy, K., & Mian, S. (1999). *Modelling gene expression data using dynamic Bayesian networks* (Tech. rep.). Berkeley, CA: Computer Science Division, University of California.
53. Nagasaki, M., Doi, A., Matsuno, H., & Miyano, S. (2003). Genomic Object Net: I. a platform for modeling and simulating biopathways. *Applied Bioinformatics*, *2*, 181–184.
54. Nagasaki, M., Yamaguchi, R., Yoshida, R., Imoto, S., Doi, A., Tamada, Y., Matsuno, H., Miyano, S., & Higuchi, T. (2006). Genomic data assimilation for estimating hybrid functional petri net from time-course gene expression data. *Genome Informatics*, *17*(1), 46–61.
55. Ong, I. M., Glasner, J. D., & Page, D. (2002). Modelling regulatory pathways in *E. coli* from time series expression profiles. *Bioinformatics*, *18*(ISMB2002), S241–S248.
56. Ott, S., Imoto, S., & Miyano, S. (2004). Finding optimal models for small gene networks. *Pacific Symposium on Biocomputing*, *9*, 557–567.
57. Pearce, D. J., & Kelly, P. H. (2006). A dynamic topological sort algorithm for directed acyclic graphs. *ACM Journal of Experimental Algorithmics*, *11*, 1.7.

58. Pe’er, D., Regev, A., Elidan, G., & Friedman, N. (2001). Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, *17*(Suppl. 1, ISMB 2001), 215–224.
59. Perrier, E., Imoto, S., & Miyano, S. (2008). Finding optimal Bayesian network given a superstructure. *Journal of Machine Learning Research*, *9*, 2251–2286.
60. Pilpel, Y., Sudarsanam, P., & Church, G. M. (2001). Identifying regulatory networks by combinatorial analysis of promoter elements. *Nature Genetics*, *29*, 153–159.
61. Savoie, C. J., Aburatani, S., Watanabe, S., Eguchi, Y., Muta, S., Imoto, S., Miyano, S., Kuhara, S., & Tashiro, K. (2003). Use of gene networks from full genome microarray libraries to identify functionally relevant drug-affected genes and gene regulation cascades. *DNA Research*, *10*, 19–25.
62. Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*, 461–464.
63. SCPD. http://cgsigma.cshl.org/jian/
64. Segal, E., Barash, Y., Simon, I., Friedman, N., & Koller, D. (2002). From promoter sequence to expression: a probabilistic framework. In *Proceedings of the 6th annual international conference on research in computational molecular biology (RECOMB2002)* (pp. 263–272).
65. Segal, E., Wang, H., & Koller, D. (2003). Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics*, *19*(ISMB2003), i264–i272.
66. Segal, E., Yelensky, R., & Koller, D. (2003). Genome-wide discovery of transcriptional modules from DNA sequence and gene expression. *Bioinformatics*, *19*(ISMB2003), i273–i282.
67. Shimamura, T., Imoto, S., Yamaguchi, R., Fujita, A., Nagasaki, M., & Miyano, S. (2009). Recursive elastic net for inferring large-scale gene networks from time course microarray data. *BMC Systems Biology*, *3*, 41.
68. Shimamura, T., Yamaguchi, R., Imoto, S., & Miyano, S. (2007). Weighted lasso in graphical Gaussian modeling for large gene network estimation based on microarray data. *Genome Informatics*, *19*, 142–153.
69. Shipley, J. M., & Waxman, D. J. (2003). Down-regulation of STAT5b transcriptional activity by ligand-activated peroxisome proliferator-activated receptor (PPAR) alpha and PPARgamma. *Molecular Pharmacology*, *64*, 355–364.
70. Shmulevich, I., Dougherty, E. R., Kim, S., & Zhang, W. (2002). Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, *18*, 261–274.
71. Staels, B., Maes, M., & Zambon, A. (2008). Fibrates and future PPARα agonists in the treatment of cardiovascular disease. *Nature Clinical Practice Cardiovascular Medicine*, *5*, 542–553
72. Tamada, Y., Araki, H., Imoto, S., Nagasaki, M., Doi, A., Nakanishi, Y., Tomiyasu, Y., Yasuda, K., Dunmore, B., Sanders, D., Humphries, S., Print, C., Charnock-Jones, D. S., Sanders, D., Tashiro, K., Kuhara, S., & Miyano, S. (2009). Unraveling dynamic activities of autocrine pathways that control drug-response transcriptome networks. *Pacific Symposium on Biocomputing*, *14*, 251–263.
73. Tamada, Y., Imoto, S., Tashiro, K., Kuhara, S., & Miyano, S. (2005). Identifying drug active pathways from gene networks estimated by gene expression data. *Genome Informatics*, *16*(1), 182–191.
74. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso, *Journal of Royal Statistical Society B*, *58*, 267–288.
75. Tinerey, L., & Kadane, J. B. (1996). Accurate approximations for posterior moments and marginal densities. *Journal of American Statistical Association*, *81*, 82–86.
76. Toh, H., & Horimoto, K. (2002). Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling. *Bioinformatics*, *18*, 287–297.
77. Tsamardinos, I., Brown, L. E., & Aliferi, C. F. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, *65*, 31–78.
78. Tusher, V. G., Tibshirani, R., & Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America*, *98*, 5116–5121.

79. Yamaguchi, R.,Yoshida, R., Imoto, S., Higuchi, T., & Miyano, S. (2007). Finding module-based gene networks with state-space models – Mining high-dimensional and short time-course gene expression data. *IEEE Signal Processing Magazine*, *24*, 37–46.
80. Yamamoto, T., Nishizaki, I., Nukada, T., Kamegaya, E., Furuya, S., Hirabayashi, Y., Ikeda, K., Hata, H., Kobayashi, H., Sora, I., & Yamamoto, H. (2004). Functional identification of ASCT1 neutral amino acid transporter as the predominant system for the uptake of L-serine in rat neurons in primary culture. *Neuroscience Research*, *49*, 101–111.
81. YTF. http://biochemie.web.med.uni-muenchen.de/YTFD/
82. Zandbergen, F., & Plutzky, J. (2007). PPARα in atherosclerosis and inflammation. *Biochimica et Biophysica Acta*, *1771*, 972–982.
83. Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of Royal Statistical Society B*, *67*, 301–320.

Chapter 25
Cancer Systems Biology

Elana J. Fertig, Ludmila V. Danilova, and Michael F. Ochs

Abstract Cancer is a complex disease, resulting from system-wide interactions of biological processes rather than from any single underlying cause. The processes that drive all cancer development and progression have been termed the 'hallmarks of cancer'. With the growth of large-scale measurements of numerous molecular and cellular properties, a new approach, cancer systems biology, to understanding the interrelationship between the hallmarks is presently being developed. Cancer systems biology focuses on systems-level analysis and presently strives to develop novel data integration and analysis techniques to model and infer cancer biology and treatment response.

25.1 Introduction

To date, cancer biology researchers have discovered many properties of cancer cells, genetic mutations in cancer, and interactions between cancer cells and components of the host organism [66]. Rather than finding a single cause, this categorization has revealed that complex interactions between cancer inducing and retarding mechanisms underlie cancer progression [73, 87, 119]. In the absence of clear targetable causes of disease, therapeutic development has lagged. As a result, cancer mortality

M.F. Ochs (✉)
Division of Oncology Biostatistics and Bioinformatics, Johns Hopkins University,
550 N Broadway, Suite 1103, Baltimore, MD 21205 USA
e-mail: mfo@jhu.edu

L.V. Danilova
Division of Oncology Biostatistics and Bioinformatics, Johns Hopkins University,
550 N Broadway, Suite 1103, Baltimore, MD 21205 USA
e-mail: ldanilo1@jhmi.edu

E.J. Fertig
Division of Oncology Biostatistics and Bioinformatics, Johns Hopkins University,
550 N Broadway, Suite 1103, Baltimore, MD 21205 USA
e-mail: ejfertig@jhmi.edu

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_25,
© Springer-Verlag Berlin Heidelberg 2011

rates have remained relatively constant over the last 50 years [90], leaving cancer as one of the five leading causes of death in the United States [82].

Biological mechanisms explain the emergence of processes that underlie malignant cancer, including most notably cell-stroma and cell-cell interactions [29, 81, 131], signaling networks [62, 94], reprogramming of expression [16, 47], genetic instability [40, 115, 136, 175], and stem cells [45, 134]. In the new era of bioinformatics, several measurement platforms have been developed and implemented to understand these cancer processes in isolation. Nonetheless, cancer is a fundamentally complex disease affecting the entire biological host system, similar to the processes in aging [50]. Moreover, the aggressiveness and metastasis of a specific cancer depend strongly on the nature of its interaction with the host organism [22,176]. As a result, seemingly eradicated tumors may later grow back aggressively after initial success from localized treatments.

A new systems biology approach is emerging to complement traditional reductionist techniques employed in cancer biology. This field uses statistical techniques and predictive mathematical models to integrate measurements from and biological understanding of individual processes in cancer. Ideally, by considering measurements and biology from the entire system, these algorithms will quantitatively and accurately test and generate hypotheses about cancer development [9, 56, 107]. Moreover, this quantitative study will ideally implicate the underlying biological processes in the development and maintenance of the malignant phenotype to aid in improving prediction of prognosis and treatment strategies [73, 87].

The statistical challenges of cancer systems biology are daunting. The organism, even the individual cancer cell, is immensely complex and poorly elucidated in comparison to a physical or chemical system. Models must address multicellular interactions and distal signals from other systems in the organism and environment. Physical cellular models are correspondingly incomplete, capturing only a fraction of the true complexity. Moreover, the cellular components modeled can never be truly isolated from the rest of the system, even in a well controlled experiment. Statistical models must, therefore, deal with this uncertainty.

While the complexity of the underlying system is extreme, we are beginning to acquire large data sets that offer opportunities for disentangling at least some of that complexity. Beginning with microarrays in the mid-1990s [109, 146], a series of technologies have been developed to allow genome-wide measurement of the dynamic molecular components of cells. We can now routinely measure numerous molecular components of cancer and normal cells. Genetic variations across genomes can be measured with SNP-chips [100], and genome-wide association studies (GWAS) are underway in many cancers [75]. Methylation of DNA can be determined globally by methylC mass spectrometry (MS) [106] and within CpG islands using arrays [20]. Measuring transcription levels of both genes [147] and exons [179] has become routine, with large data repositories available. Protein levels are measured using MS, although this remains more difficult and is not yet genome-wide. Metabolite levels for hundreds of chemicals can now be measured using MS, and structural information can be obtained using NMR. All of these data remain noisy in a statistical sense, and this noise is often confounded with natural variation in the cell. However, while certain species of molecules can be tightly controlled, differentiating between technical and biological variance is difficult given the limited replication of data points.

Global molecular measurements are being made on different tissues, which have different levels of heterogeneity in the cells comprising them. Traditional uses of the technologies noted above require large numbers of cells, so that the measurements reflect averages over a heterogeneous population. It is now clear from a number of studies that individual cells exhibit stochastic variation in the levels of molecular components, even when those levels are tightly controlled [28,99]. At present, most measurement strategies do not capture the dynamic nature of these systems, but only provide an average over asynchronous stochastic behavior even within a single cell type. Single cell analysis is a powerful approach for cultured cells, but it is unlikely that we will see such measurements on all components of a complex, heterogeneous system like a tumor for many years.

Analysis of the data generated by these new technologies can rely on the substantial understanding of cellular systems developed over the last 150 years by reductionist techniques. This information, if integrated properly into analysis, greatly reduces the potential range in the model parameters that need to be considered in fitting the data. Inclusion of this data is done both through model construction and by use of a Bayesian framework. However, in developing any algorithm, it is important to note that systems biology currently relies on noisy data and employs simplified and error-prone models to represent the cancer system.

In the next section (Sect. 25.2), we describe the biological processes involved in cancer, to lay the foundation for understanding the role of statistical analysis. The description of these processes involves a substantial vocabulary, described in the glossary in Table 25.1. Building on this highly simplified biological picture, we will elaborate on the statistical techniques used for systems biology in later sections.

25.2 Cancer Etiology

25.2.1 The Hallmarks of Cancer

Cancer is initiated throughout the body from numerous genetic mutations. While studies have revealed mutations in more than a thousand genes, only a limited number of these can drive cancer development [63]. Moreover, rarely does the mutation of a single gene initiate cancer from normal cells [65]. Instead, groups of these mutations of oncogenes (i.e., drivers of oncogenesis) and tumor suppressors (i.e., blockers of oncogenesis) collaborate to induce the genomic instability that causes further mutations and to affect seven discrete processes that lead to aggressive tumors [65,66]. Specifically, cancer cells have self-sufficiency in growth signals, insensitivity to anti-growth signals, an ability to drive tissue invasion and metastasis, limitless replicative potential, and sustained angiogenesis [66]. Recently,

Table 25.1 Glossary

Term	Definition
Acetylation	A reaction introducing an acetyl functional group into an organic compound.
Allele	A form of a gene, which is typically found in a pair in cells containing a double genome (diploid cells).
Angiogenesis	Growth of new blood vessels.
Carcinogenesis	Process that transforms normal cells into cancer cells.
ChIP-on-chip (ChIP-chip)	Measurement technology that combines Chromatin ImmunoPrecipitation (ChIP) with microarray technology (chip) to identify binding sites of DNA-binding proteins on a genome-wide basis.
ChIP-Sequencing (ChIP-Seq)	Measurement technology that combines Chromatin ImmunoPrecipitation (ChIP) with massively parallel DNA sequencing to map global binding sites of DNA-associated proteins.
Chromatin	The combination of specific DNA-binding proteins, histones, and DNA, which can be condensed.
CpG island	Genomic regions that contain a high frequency of CG nucleotides in sequence.
DNA Methylation	Addition of a methyl group to the C nucleotide in a CG sequence of DNA that may suppress transcription of a nearby gene.
Endothelium	Thin layer of cells that line the interior surface of many internal vessels, separating interior from exterior.
Epigenetic	Describing changes in phenotype or gene expression that are inheritable but not related to the DNA sequence.
Epithelium	The layer of cells lining the inside of organs or glands and the outside of the body.
Fibroblast	A connective tissue cell that secretes components of the extracellular matrix and the stroma, and which is important in wound healing.
Flow cytometry	A measurement technique for counting and sorting cells based on their molecular properties, especially cell surface proteins.
Gene	A region of DNA that encodes information for a hereditary characteristic.
Gene expression	The process by which cells take the code in DNA and **transcribe** it into mRNA and then **translate** the mRNA into protein.
Inflammation	The response of the immune system to a stimulus from infection or wounding, resulting in a number of physiological changes.
Kinase	An enzyme that transfers phosphate groups to specific amino acid residues on proteins.
Knock-out, knock-in, knock-down	Experimental techniques to remove, insert, and reduce, respectively, expression of a gene.
Meiosis	A process of cell division resulting in each daughter cell having half the number of chromosomes.
MicroRNA (miRNA)	A small RNA molecule encoded in the DNA that targets specific mRNAs reducing expression of some genes.
Mitochondria	A cell organelle that produces energy for a cell.
Mitosis	A process of cell division resulting in each daughter cell having the full number of chromosomes.
Phenotype	Any *observable characteristic* or trait.

(*continued*)

Table 25.1 (continued)

Term	Definition
Phosphatase	An enzyme that removes phosphate groups from specific amino acid residues on proteins.
Polymerase Chain Reaction (PCR)	A technology that amplifies a piece of DNA to permit measurements requiring more than a few molecules.
RNA interference (RNAi)	The reduction of mRNA translation by complementary RNA sequences to that of the mRNA.
Reverse Transcription PCR (RT-PCR)	A technology that converts an RNA molecule to a DNA molecule and then amplifies it using PCR, which can be used for highly sensitive transcript level measurements.
Single-Nucleotide Polymorphism (SNP)	A sequence variation that occurs within a significant fraction of the population, such that a single location in the genome has two different bases in many individuals.
Small interfering RNA (siRNA)	A fabricated RNA designed to cause RNA interference and knock-down a gene in an experiment.
Telomere	A region of repetitive DNA at the end of chromosomes, which shortens with each DNA replication.
Transcription	Copying a strand of DNA into complimentary RNA to promote gene activity.
Translation	Process by which mRNA is converted into proteins.
Vogelgram	Figure representing the stages of cancer and the physiological drivers of transitions between stages.

inflammation has been recognized as an additional hallmark of cancer [97, 112]. These ‘hallmarks’ are summarized in Fig. 25.1.

The ultimate goal of cancer systems biology is to infer the malignant drivers of the biological processes present in an individual patient’s cancer. Normal cells maintain themselves by producing needed molecular components (e.g, proteins, metabolites, nucleic acids, lipids) to perform their functions. This maintenance requires expression of genes as needed according to the Central Dogma (DNA is **transcribed** to mRNA that is **translated** to protein). A subset of cells in adults divide to produce new cells. During this process, cells receive a signal to proliferate (e.g., a growth signal), and they go through the cell cycle which involves preparation for DNA synthesis (G1), replication of DNA (S), preparation for mitosis (G2), and cell separation by mitosis (M). Each division of a cell leads to shortening of the ends of the chromosomes, called telomeres, and replication is blocked once telomeres reach a minimum size. Cells are also equipped with mechanisms to block proliferation through anti-growth signals and to control cell death through cell death signals. Cell death mechanisms can trigger an irreversible self-destruction involving programmed rupture of the mitochondrial organelles in the cell. These apototic processes can also be triggered by detection of an unfixable problems, such as coding errors in the DNA. Therefore, to grow and evade cell death, cancer cells must both inappropriately activate proliferation and suppress normal cell death processes. As cancers grow, further short-circuiting of normal biology is required to induce the

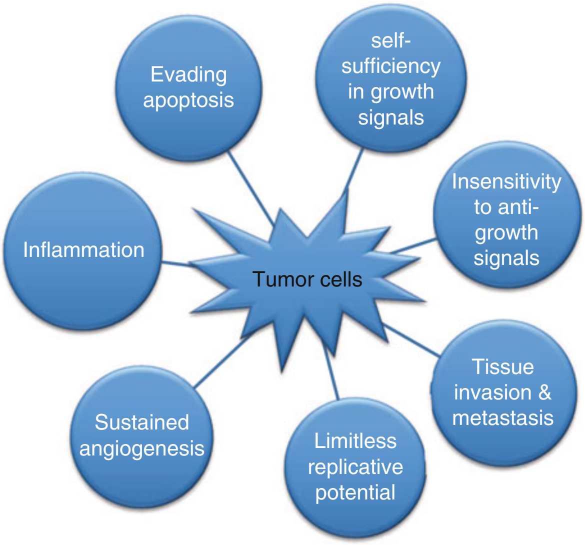

Evading apoptosis
self-sufficiency in growth signals
Insensitivity to anti-growth signals
Inflammation
Tumor cells
Tissue invasion & metastasis
Sustained angiogenesis
Limitless replicative potential

Fig. 25.1 The hallmarks of cancer. These biological processes need to be deregulated in order for a cancer to grow and metastasize, which is why cancer generally involves a series of changes (modified from [66])

biological processes that comprise the hallmarks. Algorithms inferring the sources of cancer progression may identify the mutations to the DNA or the sources of alterations in gene expression that can short-circuit normal cellular function. Similarly, algorithms that infer the effects of the hallmark processes may improve early disease detection and discover processes that promote development and metastasis of cancer cells. Therefore, knowledge of the cancer hallmarks is required to guide interpretation of statistical analyses, distinguishing key processes in the cancer system from other processes that might be changed from normal cells but that are incapable of driving carcinogenesis. Ideally, incorporation of prior knowledge on genes and processes, as in a Bayesian model, may improve inference of these processes and their underlying sources.

The first process, **evading apoptosis** (programmed cell death), involves the loss of responsiveness to signals indicating that the cell needs to eliminate itself. The master regulator of this process is the protein p53, which integrates signals from external cell fate signals and from internal quality control signals that monitor DNA stability [169]. The second process, **self-sufficiency in growth signals**, is characterized by many mutations that drive cancer. For example, a number of cancers contain mutations in growth factor receptors or in downstream activators of growth factor receptors, making them active in the absence of growth signals [6, 80]. The third process, **insensitivity to anti-growth signals**, involves sets of proteins that are designed to block activated growth signals by deactivating downstream components [86]. The fourth process, **limitless replicative potential**, often involves activation of the enzyme telomerase, which can lengthen the telomeres at the end of the chromosomes [68]. The fifth process, **sustained angiogenesis** (i.e., creation of new blood vessels), is important for solid tumors, which require nutrients to be delivered to enable them to continue to grow [51]. The sixth process, **tissue invasion and metastasis**, involves remodeling of the cellular environment and degradation of surrounding structures [126], permitting a cancer cell to leave the original site and circulate, finding a new site to begin a new tumor (i.e., metastasis). The seventh process, **inflammation** (e.g., activation of wound and infection response), allows recruitment of cells and chemicals to the site of a wound, and it is hijacked by cancer to avoid attack by the immune system and to acquire nutrients [141].

However, targeting these processes is difficult for an individual tumor, because the specific mutations are highly variable between tumors, and because targeting these processes will affect normal cells causing toxic side effects [119]. Thus, to eradicate a tumor, therapeutics must target the processes within the specific tumor cells and potentially additional cells harboring tumorigenic potential, while avoiding damaging normal cells. Therefore, one goal of analysis in cancer studies is to distinguish cancer cells from their normal counterparts. Statistical algorithms, such as class comparison, can identify underlying genetic differences between cancer and normal tissues. Clustering and pattern recognition algorithms can extend these inferences to identifying features that correlate to the hallmark processes. Integrating these features with knowledge of the biological processes underlying the cancer hallmarks (several of which are described in the remainder of this section) can implicate candidate biomarkers and therapeutic targets for individual cancers or cancer types, targeting of which will ideally have minimal impact on normal tissues.

25.2.2 Cell-Stroma and Cell-Cell Interactions

In addition to mutations, the microenvironment to which cancer cells belong can also promote tumorigenesis [5, 32, 81]. The need for a suitable microenvironment for tumor cell growth was predicted early by Paget in his *seed and soil hypothesis* [127]. It is now believed that as cancer cells develop, they co-evolve the components in the surrounding environment, together called the *stroma*. The resulting cell-cell interactions between the cancer and stroma promote tumor growth and metastasis [104]. As a result, statistical algorithms that can infer abnormalities in the stroma or in cell-stroma interactions may provide novel insights into the cancer system. Such algorithms must perform multivariate analysis to infer the different covariates that correspond to cell types and interaction terms for cell-stroma interactions.

The stroma contains fibroblasts, endothelial cells, immune cells, and the extracellular matrix that can promote tumorigenesis and metastasis [2, 104]. In normal systems, the components of the stroma facilitate normal development by secreting the extracellular matrix, scaffolding and shaping connective tissue, healing wounds, and mediating the immune response [4, 104]. However, in tumors, the complex cross-talk between cancer cells and the stroma depicted in Fig. 25.2 causes the stroma to send signals to the cancer cells that induce and sustain the hallmarks

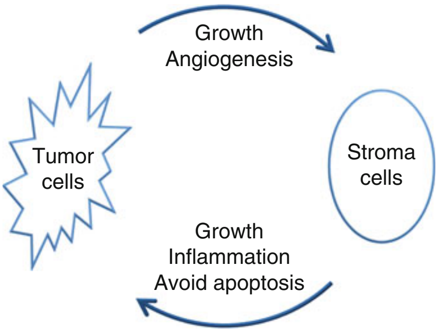


Fig. 25.2 Complex cross-talk between the tumor and surrounding stroma promote tumor development and metastasis. Tumor cells send signals to the stroma that encourages its growth and recruitment of blood vessels. Simultaneously, the stroma sends signals to the tumor encouraging its growth, inflammation, and continued survival

of cancer [60, 131]. This role of stroma immune cells in tumorigenesis is recognized in references to tumors as ‘wounds that do not heal’ [42]. The tumor growth that results from this immune response further promotes recruitment of additional stroma components and vasculature to ultimately enable metastasis into otherwise oxygen-poor (hypoxic) and nutrient-poor environments [32, 104]. As a result, traditional therapeutics that target only cancer cells may leave behind an environment that is suitable for reinvasion, and, furthermore, may trigger an immune response in the myofibroblasts of the stroma that could later encourage regrowth of the tumor [37]. Moreover, traditional cancer treatments such as chemotherapy, radiation, and surgery may promote activity of the aberrant immune cells in the tumor stroma to facilitate regrowth after treatment [37].

Although the stroma apparently promotes tumor development and metastasis, the specific sequence of events that triggers malignancy in otherwise normal cells and stroma remains unknown [5, 60]. Nonetheless, the early role of the stroma in tumorigenesis is apparent from experiments revealing that the growth and metastasis of tumor cells is limited *in vitro* without the presence of the stromal components [37] that are observed in the early metastatic process *in vivo* [37, 60, 140]. Mathematical models can be used to explore the necessary co-evolution of the tumor cells and stroma in many scenarios that cannot be observed directly in experiments due to technical and ethical limitations. Many of these models solve systems of ordinary or partial differential equations that describe the growth of populations of tumor cells and stroma components due to hypothesized interactions. For example, systems of equations can model the effects of increased nutrients in the tumor due to recruitment of blood vessels in the stroma [10, 55, 58], growth signals propagated between the stroma and tumor cells [165], and metastatic breaches [9, 10, 122, 135]. As a result, these predictive models can provide powerful and non-invasive *in silico* tools to test hypotheses about the role of the tumor microenvironment in cancer growth and test the efficacy of different targets in that environment.

While the mechanisms underlying tumor cell and stroma development are still being explored, several researchers are pursuing promising therapeutics that directly target the microenvironment in order to kill cancer cells and prevent metastasis [2, 5, 104]. These studies attempt to identify distinguishing malignant stroma features by comparing gene expression measurements from tumor stroma to a control population with cDNA libraries, Serial Analysis of Gene Expression (SAGE), and microarray measurements. The resulting gene expression measurements are compared to look for differences between gene activity (called differential expression) in the measured population types [2, 13]. In class comparison algorithms, gene expression from stromal cells is compared to that from non-stromal cells [71] (see Sect. 25.3.4.1). To avoid inadvertently targeting the normal stroma components in healthy tissue, therapies must target only those stroma components that behave aberrantly in malignant stroma [104]. Therefore, class comparison algorithms may yield more effective targets if applied to distinguish gene expression in tumor stroma from normal stroma (e.g., [33, 181]). Clustering algorithms (see Sect. 25.3.4.2) may also be applied to these datasets to distinguish gene expression in normal stroma from that in tumor stroma [11, 13, 49]. However, as described above, it is likely that the differences in functionality and expression in tumor stroma from their normal counterparts results in response to signals received from the neighboring tumor cells. Therefore, algorithms used to infer cell signaling processes (Sect. 25.2.3) may be most adept at inferring both the underlying processes in and the sources of aberrant behavior in tumor stroma [77, 138].

25.2.3 Signaling Networks

As described in the previous section, cellular communication is crucial to both beneficial and malignant interactions between a cell and its environment. As a result, statistical techniques which infer the biological species which mediate this communication and the subsequent phenotypic effects are critical to understanding the cancer system. Many cellular processes, including notably those induced through cell-stroma interactions, are controlled by reactions between intracellular proteins (particularly kinases and phosphotases) that occur in a specified sequence in response to triggers from the cell's external environment and internal state. Each of these reactions is known as a *signaling reaction* and the sequence in which they occur is referred to as a *signaling pathway*. Together, collections of signaling pathways within a cell form a *signaling network*. In cancer, activity in these signaling networks plays a key role in tumor development and metastasis [18, 111]. For example, signaling networks facilitate the interactions between tumor cells and the tumor stroma described in Sect. 25.2.2. In this example, the growth signals secreted by the tumor stroma trigger signaling pathways in the tumor cell that facilitate cell division and hinder apoptosis, while the corresponding signals sent from the tumor cell trigger pathways that promote angiogenesis and immune response in the stroma.

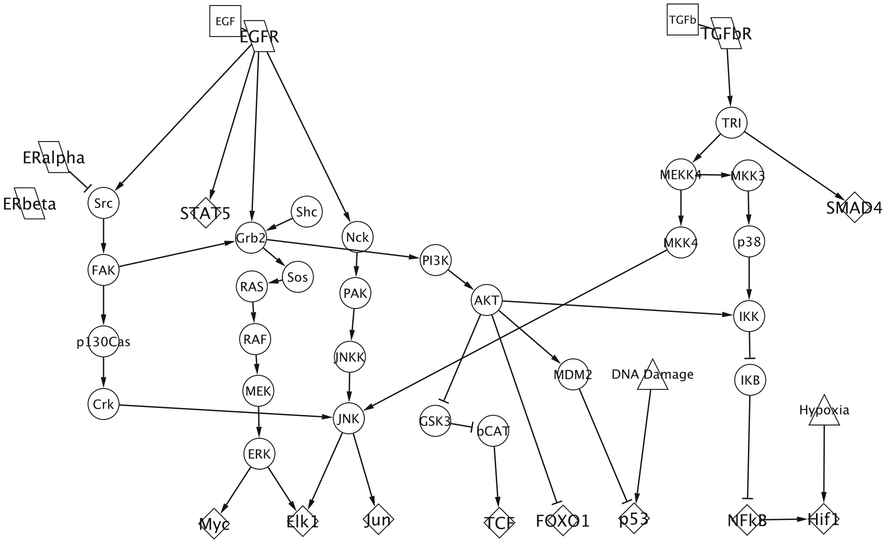


Fig. 25.3 Graphical representation of a signaling network relevant to processes in breast tumors. The *top parallelograms* represent receptors in the cell membrane, while the *bottom diamonds* represent transcription factors in the nucleus. The growth factors (*squares*) bind the receptors causing them to communicate with the signaling proteins in the cytoplasm (represented by the *circles*). Each *straight arrow* in the cytoplasm represents an activation of the target protein by the origin protein, and each T the repression of the target protein. *Triangles* represent states in the cell, such as hypoxia, which can induce signaling, often through unknown intermediaries

Figure 25.3 depicts an example of a signaling network that enables tumor cells to respond to internal and external triggers, such as the signals from the tumor stroma. In this network, extracellular signals shown as squares are processed by the receptors shown in the figure as parallelograms. These receptors signal to the downstream proteins (shown as circles) in the cytosol of the cell by inducing post-translational modifications, typically phosphorylation at specific amino acid residues. The signals can be activating or repressing, and their final targets are often transcription factors (shown as diamonds), leading to reprogramming of gene expression (see Sect. 25.2.4). The networks often have feedback loops directly or through gene expression, feed-forward loops, and substantial cross-talk that enable complex reactions to stimuli with a limited number of signaling components [15, 85]. In addition, while core pathways have been deduced, there are numerous proteins known to play a role in signaling that are not yet reflected in core diagrams (such as Fig. 25.3). Signaling pathways often regulate cell growth, proliferation, death, and adhesion, so therapeutics that modify these signaling processes could directly target the source of cancer progression for each patient [150, 167].

Ideally, signaling structure and activity could be inferred from direct measurements of the population and activity of each protein within a cell. However, while informative, such direct proteomic measures are restricted to a few proteins *in vivo* at limited times during tumorigenesis [15, 153]. Nonetheless, the structure and activity in signaling networks can be inferred from numerous indirect measurements. Often, the signals passed along pathways promote gene expression to facilitate the desired biological process. Therefore, global gene expression measurements from microarrays indicate active pathways in measured cells [21]. Chromatin ImmunoPrecipitation-on-Chip (ChIP-Chip) experiments use microarray technology to identify the locations on the genome to which proteins that control gene expression (*transcription factors*) bind. Integrating these measurements with global gene expression can indicate the interactions of these transcription factors with pathway activity, elucidating the activity in specific pathways and the effects of cross-talk [128, 143, 166]. Similarly, classes of RNAs, including siRNAs [128, 139, 142, 167], can *knock-down* specific proteins in signaling networks. Therefore, targeted experiments with these siRNAs can further probe the implications of network structure on signaling activity. Moreover, several genetic variations in an individual are likewise linked to specific modifications in signaling activity. As a result, expression measurements from individuals with genetic variations measured by SNPs can further provide indirect indications of the signaling activity in that individual [84, 130].

Because some transcript level changes derive from signaling activity, the structure and activity in signaling networks can be inferred with statistical techniques (reviewed in [30]). Often, algorithms based upon Bayesian networks can infer the structure of the signaling network that best fits the gene expression measurements (e.g., [26, 70, 139]). These algorithms can utilize prior distributions that encode structural information from additional sources [19, 120, 160, 178]. While successful at inferring the structure of networks with few connections and proteins, these techniques cannot account for feedback loops [139] and are subject to overfitting when used to infer the structure of moderately-sized networks [120].

The structure of several core networks have been established and are available in curated databases (e.g., [174]). These databases often contain lists of protein-protein interactions along specific signaling pathways that have been reported in numerous independent studies and verified by the database curator. While these databases may still have errors and missing interactions, they ideally provide more accurate estimates of the structure of signaling networks than that inferred from any single gene expression experiment. Several statistical algorithms utilize this established structure of signaling networks to infer the signaling processes active along specific pathways in tumor samples in order to identify optimal molecular targets for treating these cancers. These techniques include standard statistical analyses of differential expression [148], clustering [137], and pattern recognition [21], including use of transcription factor target information [96]. While standard in machine learning, many of these algorithms are limited in their utility for biological inference because they cannot fit the large number of parameters [89], account for feedback loops [139], or incorporate reuse of genes for multiple cellular functions [118]. Modifications of these approaches include use of non-negative matrix factorization techniques to identify overlapping groups of coregulated genes [91, 118] and supervised learning based on sets of genes derived from the networks [96, 159].

Computational modeling using the structure of the signaling networks as the basis for reactions in differential equation-based models is also an area of active research [113, 139, 144, 170]. Ideally, these models will one day predict signaling processes that are active in a cancer and the implications of different targeted treatment strategies. However, these models are currently limited to modeling signaling in simpler organisms than humans and their cancers because of needed but unmeasurable model parameters and the complex dynamics of the multicellular system [73, 89]. Therefore, while protein signaling shows the potential to target cancer progression at its source, quantitative inference of signaling activity in a tumor requires a systems approach to integrate the information provided from all sources of indirect measurement of the biological processes coupled with predictive mathematical models [53, 73, 74].

25.2.4 Gene Expression and Epigenetics

The specific genetic mutations that induce cancer vary widely across patients. However, similar changes in expression of specific genes are commonly observed during tumor growth and metastasis, suggesting a need for similar molecular components [47]. Although not encoded in the DNA sequence, some changes in gene expression can be inherited by cancer cells to favor disease development [16, 48], potentially facilitating later genetic mutations [16,47]. Such non-genomic inheritance is known as *epigenetic* inheritance. For example, the gene PTEN is epigenetically silenced in several human cancers, which leads to increased AKT activity and loss of tumor suppression [52, 117, 151, 156]. In cancer, epigenetic effects include heritable markers that promote or repress gene activation, often achieved by binding of methyl groups to target sites, known as *CpG islands*, upstream of a gene on the DNA [16, 47, 48]. Loss of these methyl groups, *hypomethylation*, can promote gene expression, while *hypermethylation* can silence expression. Typically methylation is used specifically to silence genes following organism development, in order to shutdown growth processes needed only in embryogenesis. As such, hypomethylation can reinitiate these processes, which can lead to the uncontrolled growth seen in cancer. Alternatively, hypermethylation of a needed tumor suppressor, such as PTEN, can remove checks on such growth. Cancers may also modify gene expression by altering the compressed structure of the DNA (i.e., *chromatin*), which sequesters genes away from transcription factors. In addition, expression of microRNAs (miRNAs) can similarly reduce the production of protein by targeting mature mRNAs for destruction prior to translation [35]. Identifying the sources of expression changes may improve clinical detection and diagnostics. For instance, improper hypermethylation changes may be reversed by chemical compounds, providing powerful cancer therapeutics [72].

Changes in transcript levels will ultimately lead to a different complement of proteins being produced by the cell. The most straightforward way to identify these effects would be to measure the protein states and levels. However, such measurements are not presently feasible on a genome-wide basis. Alternatively, microarrays (and now exon arrays) can measure transcript levels (and alternative splicing). However, excluding highly expressed proteins, correlation can be low between mRNA levels and protein levels [64, 168]. In general, it seems highly likely that creation of increased levels of mRNAs whose proteins all work together in a biological process represents upregulation of that process. As such, one approach to the analysis of cellular reprogramming is to focus on gene set enhancement of biological process gene ontology terms.

Using microarray measurements to identify the genes expressed at higher or lower levels relative to normal tissues could improve predictions of prognosis. For example, class comparison algorithms are employed to infer specific genes that distinguish normal and cancer samples. However, while microarrays measure global gene expression, at most they can provide information about correlation to specific malignancies when using clustering or pattern finding algorithms. Thus, cancer systems biology algorithms which rely on microarrays alone are insufficient to identify the cause of changes in gene transcript levels.

Identifying the causes of changes in gene expression in tumor samples will require integration of transcript measurements from microarrays with measurements from additional platforms. DNA methylation arrays indicate the points on the genome to which methyl groups are bound, identifying sites for hypo- and hypermethylation of the DNA [54, 116]. Methylation arrays are first analyzed to determine the specific genes that are routinely hyper- and hypomethylated [24, 79]. ChIP-chip and ChIP-seq measurements provide additional information about interactions between transcription factors and DNA, further suggesting potential locations of hypo- and hypermethylation of the DNA [24, 54, 166]. Finally, miRNAs can also be measured using specialized microarrays and these measurements can be correlated with transcript levels of mRNAs targeted for degradation [1]. Alternatively, quantitative real time polymerase chain reaction (qPCR) can directly detect and quantify activity in specific miRNAs known to reprogram expression in cancers [1]. Together, these data sets can be used to identify mechanisms for malignant expression modifications through algorithms discussed in Sect. 25.3.3. For example, analysis can propose driver biological mechanisms by seeking methylation, miRNA, or genetic alterations that correlate with gene expression changes identified using class comparison, clustering or pattern recognition algorithms [155].

25.2.5 Genetic Instability and 'Vogelgrams'

The biological processes that induce carcinogenesis in normal cells often decrease the fidelity in cell replication, causing cancer cells to become genetically unstable [23, 36, 136]. This genetic instability subjects cancer cells to numerous mutations upon replication. As a result, many statistical algorithms used to infer mutations distinguishing normal and cancer samples will find many mutations across studies, most of which cannot drive oncogenesis [63]. Currently, competing theories are invoked to explain the inception of genetic instability in cancer, including microenvironment-tumor interactions [67] (Sect. 25.2.2), disruption of DNA repair by growth preferring mutations [110], improper signaling and expression [23] (Sects. 25.2.3 and 25.2.4), and microRNA activity [36] (Sect. 25.2.4).

Ideally, identifying driver genes could implicate the source of cancer progression, and thus suggest a therapeutic strategy. Although it is challenging to distinguish driver from passenger genes in these subsets, changes in candidate driver genes likely confer a change in the information encoded in the DNA, for example resulting in a change in an amino acid, or in a promoter where a transcription factor binds. Comparisons of such mutations in several patients have implicated numerous driver genes in each stage of cancer development [63]. However, comprehensive studies integrating genetic and transcript measurements from numerous tumor samples have shown that the driver mutations may primarily occur within a limited number of core signaling pathways [84, 130, 163, 173], as expected from the hallmarks.

To systematically track disease progression, figures called *Vogelgrams* link the sequence of phenotypic changes in an evolving cancer cell to the specific driver mutations responsible for each change [46]. An example of a generalized Vogelgram is given in Fig. 25.4, where the types of genetic changes, rather than specific mutations, are provided for a single cancer type. Comparison with Fig. 25.1 shows how the hallmarks of cancer tend to be reflected in the transitions in a Vogelgram.

The sequence of mutations in cancer has been used to abstract the mutation processes in stochastic and differential equation models that predict the amount of time and number of distinct mutations needed for cancer development [17, 95, 110,

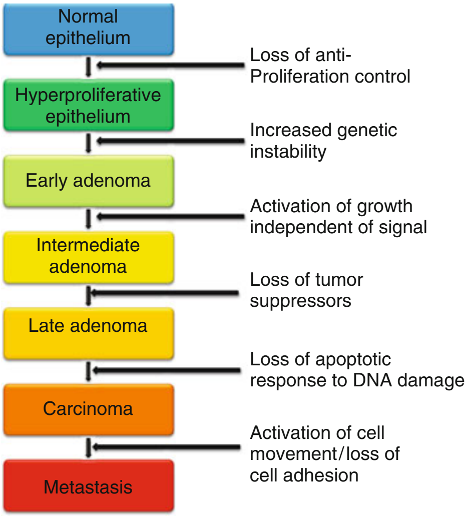

Normal epithelium
Loss of anti-Proliferation control
Hyperproliferative epithelium
Increased genetic instability
Early adenoma
Activation of growth independent of signal
Intermediate adenoma
Loss of tumor suppressors
Late adenoma
Loss of apoptotic response to DNA damage
Carcinoma
Activation of cell movement/loss of cell adhesion
Metastasis

Fig. 25.4 Generalized from the Vogelstein model of colorectal cancer progression (from [92]) that extends the Knudson two-hit model [93]. The progression of the cancer from a benign neoplasm to a carcinoma involves multiple genetic mutations involving different genes and potentially different sequences in various cancers. These models have been refined and can be constructed for many different types of cancer

115, 133]. These models have provided significant evidence that genetic instability is an early precursor to tumor formation, possibly even responsible for initiating tumorigenesis [115]. Another view is that it is errors in cell signaling or methylation changes in DNA that drive the emergence of genetic instability. This latter hypothesis can be explored by identifying the underlying drivers of biological modifications in the cancer system such as those suggested in Sect. 25.2.4. Specifically, techniques that can analyze integrated SNP, copy number, methylation, miRNA, and expression alterations in cancer samples should be particularly adept at inferring the underlying drivers of genetic instability.

25.2.6 Stem Cells

The complexity of the biological processes in tumorigenesis make improbable the ability of multiple heterogeneous cell types to simultaneously become cancerous. However, tumors are composed of such heterogeneous cells, which could be explained if, like tissues during development, tumors develop hierarchically from a common stem cell [38, 108, 177]. Like cancer cells, stem cells evade apoptosis and have unlimited replicative potential, thereby requiring fewer mutations to gain all necessary cancer functions than normal differentiated cells [45, 134]. Therefore, resistance of stem cells to therapeutics, perhaps due to the fact that stem cells can lie dormant and most cancer treatments rely on cells growing, could cause seemingly eradicated tumors to regrow [45, 108, 154]. Moreover, if stem cells acquired genetic instability, they could easily mutate in response to treatment to confer a growth advantage and resistance to treatment [98]. Thus, under the stem cell hypothesis, therapeutics must target cancer stem cells in addition to the tumor cells [45, 154]. The dynamics of tumorigenesis and the response to therapeutics arising from stem cells is supported by mathematical models [114].

In contrast to the stem cell model, a second model of cancer involves clonal evolution, where one differentiated cell acquires attributes that permit uncontrolled growth and it subsequently comes to dominate the other cell types. Both models have aspects that appear to agree with experimental results [172]. For instance, in many cancers multiple cell types appear tumorigenic in xenograft models, suggesting differentiated cells have tumorigenic potential. However, certain cell types identified by specific cell surface markers appear to have increased and sometimes dominant tumorigenic potential, as suggested in the cancer stem cell model and first identified in breast cancer [3]. However, the current measurements and models cannot distinguish true cancer stem cells from normal cells that have mutated under the pressures of the cancer to evolve stem-like properties [103, 152, 171], as our stem-cell markers are not highly specific.

Activity in particular transcription factors is one indirect marker of cancer stem cells. As a result, differential expression analysis between cancer and normal cells focused on inferring transcription factor activity may implicate stem-like behavior in measured tumors [125]. However, such inference cannot distinguish biological modifications that induce stem-like activity and a predominance of cancer stem cells in the tumor. Integrative analysis of measurements from multiple platforms may infer drivers of this stem-like behavior through genetic or epigenetic modifications, but may not distinguish between modifications that caused normal stem cells to attain carcinogenic properties and changes that permitted cancer cells to attain stem-like properties.

25.3 Towards a Systems Biology of Cancer

The previous section (Sect. 25.2) described several biological mechanisms that underlie cancer. Because of the complexity of cancer, measurements from any single platform have intrinsic limitations in representing the entire cancer system [56]. As a result, numerous measurement platforms have been devised to gather data from cancer samples. The corresponding analysis algorithms in cancer systems biology have several common elements, including notably distinguishing cancer and normal samples and identifying clusters or patterns resulting from regulation by common biological drivers. While several of these inferences can be made with standard statistical algorithms, novel statistical techniques are constantly emerging to infer the biological processes responsible for inducing and maintaining cancer in an individual from these measurements.

In the remainder of this section, we describe standard statistical models that have emerged to identify the complex biological processes in the cancer system described in Sect. 25.2. The evolution of statistical models in systems biology can be predicted based on their development in response to the first systems-level measurements to emerge in biology, the gene expression microarray. These methods include ‘preprocessing’ necessary to remove technical artifacts from the data [78] (Sect. 25.3.2); data integration to bring together the different types of data (e.g., mRNA, DNA, proteomic measurements) and annotate them (e.g., gene ontology) to allow improved analysis (Sect. 25.3.3); and analysis to identify statistically significant variations (Sect. 25.3.4). Many algorithms for these analyses are implemented within the R Project for Statistical Computing using Bioconductor [57], and many other tools have been created though often they rely on their own data formats.

25.3.1 Data for Systems Biology

Section 25.2 included discussion of several measurement technologies used to elucidate the biological processes in cancer (summarized in Table 25.2). As one of the first global measures of biological systems, microarrays provide the most common data source in experiments probing the processes in tumor development, maintenance, and metastasis. Moreover, because many publishers require all measurements from microarray experiments to be deposited in the Gene Expression Omnibus database (GEO) [14] or in ArrayExpress [129] upon manuscript submission, there is an abundance of public data available for the development, testing, and implementation of statistical algorithms. However, it is important to remember that gene expression is technically the production of a protein from the DNA gene encoding its sequence, even though the term is often used to refer to transcripts measured by a microarray (see Table 25.1).

Statistical analysis for cancer systems biology must evolve to use measurements that can track the full complement of cellular molecular components, including DNA sequences for genes and non-coding regions (e.g., promoters, enhancers, etc.), RNA variations from alternative splicing, miRNAs and their targets, proteins including variants and their structures, and metabolites using the additional measurement platforms described in Table 25.2. In addition, it is important to carefully describe and track phenotypes, which will best be done with some type of ontology or controlled vocabulary [25].

One complication that enters with functional genomics is that context (e.g., cell type, disease state) and time (e.g., time following stimulation, time after treatment) have an impact on the measured data, which contrasts with DNA sequences, at least for typical cells. For RNA and proteins, the actual production and lifetimes of the molecules vary widely and these variations impact phenotype. A more complete understanding of the biological processes underlying cancer requires integrating measurements from multiple platforms focused on different molecular forms (e.g., DNA, RNA, protein, pathway, complex) and cells (e.g., cancer cells, stromal cells, circulating immune cells) from an individual. As a result, new databases with

Table 25.2 A snapshot of measurement technologies for cancer systems biology

Technology	Target	Uses
SNP-chip	SNP variants	Allelic variation
DNA sequencing	DNA variations	Gene mutation, allelic variation
MethylC MS	Methylation sites of DNA	Hypermethylation, gene expression
Methylation arrays	CpG methylation	Hypermethylation, gene expression
Microarrays	mRNA concentration	Gene expression
RT-PCR	mRNA concentration	Gene expression
Exon-Chip	mRNA at exon level	Gene expression, alternative splicing
SAGE	mRNA	Gene expression
ChIP-on-chip, ChIP-Seq	Protein binding to DNA	Transcription factor targeting, gene expression
miRNA arrays	miRNA levels	Gene expression
Mass Spectrometry	Protein concentration, metabolite concentration	Signaling activity, gene expression
Nuclear magnetic resonance spectroscopy	Metabolite concentration	Metabolic flux, enzyme activity
Flow cytometry	Proteins on specific cells	Signaling activity, cell concentrations, heterogeneity

multi-platform measurements of individual tumor samples are being compiled and made publicly available (e.g., [163]).

A second complication is the fact that unmeasured covariates will always play an important role in biological studies, as the systems are too complex to obtain complete coverage in measurements. In addition, as the discovery of miRNAs made clear, there remain unrecognized active biological components that we neither measure nor model. This makes identification of causative mutations or events a two-stage process. Statistical and computational approaches must make predictions from the available data, but these predictions must be verified by controlled biological experiments to show that the identified mutation is causative and not merely correlative.

25.3.2 Data Preprocessing

Throughout the remainder of this chapter, we will adopt the notation that measurements are stored in an $m \times n$ matrix $\mathbf{X}$ in which each row represents a biological entity (e.g., a gene, protein, etc.) and each column represents a sample or experiment. Ideally, some rows of $\mathbf{X}$ vary across columns due to differences in the sample or experiments, called *interesting variations*. However, variation often arises from technical artifacts in the data processing, called *obscuring variations* (discussed in [69]). As a result, *preprocessing* techniques must be implemented to remove the technical artifacts before performing an analysis on any biological measurement.

Often, the raw measurements will have dramatically different distributions across rows (biological factors) or columns (sample or experiment). However, in order to compare variations across samples, each column in $\mathbf{X}$ must have measurements on the same scale, but the technologies tend to give only relative measurements. Therefore, many preprocessing techniques strive to scale the measurements to enable comparison across columns. Many of these techniques assume that if the measurements were not subject to obscuring variation, they would have the same signal distribution across rows. Currently, the most widely used preprocessing techniques of this variety are applied to microarray data, *robust multi-array average* (RMA) [78] and DNA-chip analyzer (dChip) [145]. Similar techniques are under development for SNP-chips, miRNA arrays, methylation arrays, proteomics techniques, and metabolomics measurements.

While normalization techniques such as RMA correctly put the columns of $\mathbf{X}$ on a comparable scale, technical artifacts may remain to obscure the interesting variations. For example, even after normalization, significant variations in some rows of $\mathbf{X}$ are often observed across columns that are distinguished only by technical factors (for example, by array, processing date, technician, etc.) called *batch effects* (e.g., [121, 180]). Several groups have corrected for such batch effects by applying linear models with covariates that represent the technical variables [44, 83]. However, because these factors are unknown *a priori*, other groups have developed algorithms based upon the singular value decomposition to simultaneously estimate the covariates and correct for the batch effect from the set of measurements [102, 121].

Nonetheless, it is important to note that no algorithm can correct for batch effects if the technical covariates match the interesting covariates. For example, if a study processed all cancer samples on one day and all normal samples on another, it will always be impossible to distinguish variations due to disease status from those due to lab differences [12]. Therefore, care should be taken when designing a study or selecting a data set to randomize the processing of samples.

25.3.3 Data Integration

Because biological systems are extremely complex, measurements from many platforms should be analyzed to capture the biological processes responsible for cancer [56]. For analysis, the data must be properly integrated into a coherent data set. Once integrated, analysis algorithms must consider the different range, uncertainties, and biases in the measured data [76], in addition to the biological relationships between the different types of data.

Data integration is often achieved by formulating a score or probability of specific biological events (e.g., genetic mutations, protein interactions, etc.) based upon the multiple measurements and then applying existing algorithms to these unified scores to infer the underlying processes inherent in these measurements [76,84,130]. Alternatively, information from additional measurement platforms can be directly encoded in the analysis algorithm applied to a global measurement data set, such as microarray measurements. For example, clustering algorithms can incorporate biological knowledge from multiple data sets to initiate the clusters proposed or to define the distance function used to separate clusters [132, 149, 182]. Similarly, this biological information can be encoded as prior distributions in Bayesian algorithms, such as in Bayesian networks used to infer structure and activity of signaling networks [19, 120] or in algorithms leveraging known transcriptional regulation [96]. This inferred structure can then be used in further analysis to identify processes occurring along specific pathways [21, 163].

Such integrated analysis of multi-platform measurements will ideally improve inference of the underlying cancer system to facilitate the formulation of novel biological hypotheses and predictions. Moreover, integrating this wealth of measurements with predictive models will further refine both sources of biological information. Ideally, integrated algorithms will accurately estimate and predict the biological processes underlying a patient's cancer, and therefore, improve predictions of the prognosis and impact of treatment strategies.

25.3.4 Data Analysis

Once the measurements have been preprocessed and integrated, they can be analyzed to infer the biological factors that underlie cancer development, maintenance, and metastasis described in Sect. 25.2. The analysis algorithms typically strive to infer markers that distinguish normal and cancer samples and to identify the mechanisms which induce these differences. The algorithms can be divided into class comparison algorithms (Sect. 25.3.4.1), clustering algorithms (Sect. 25.3.4.2), and pattern recognition algorithms (Sect. 25.3.4.3). We focus here on algorithms applied to microarray data, as most developments have occurred in this area. These algorithms can be used on integrated data as well, however algorithms that address the differences in molecular components will have greater impact to identify malignant drivers and, thus, candidate treatment targets for individual cancers.

25.3.4.1 Class Comparison

Class comparison or *outcome-related gene finding* algorithms estimate differences between two (e.g., cancer and normal) or more (e.g., cancer types) classes [41]. For example, these algorithms are often applied to microarray data to infer genes whose expression differences distinguish the phenotypic classes. Biological processes or gene expression levels inferred to differ are often used as disease markers. Many standard inferential analysis techniques are used to identify such differences, including t-tests, ANOVA, and regression [7], providing subsets of biological molecules that are correlated to the disease system, and which should be explored by further assays and analyses to identify drivers of carcinogenesis. For example, if the class comparison algorithms identify genes that are commonly expressed in stem cells, experiments can generate additional data to determine if the measured cells indeed have stem-like properties, as discussed in Sect. 25.2.6.

Class comparison algorithms are typically implemented in two steps. First, a metric is established to rank the rows of **X** (often genes). This metric must incorporate the variability in each row, not merely the magnitude of the difference between the classes. Therefore, these algorithms often use metrics based upon the t-statistic for sorting [7, 13, 34]. In the second step, the top rows of this list above a threshold are retained as having a significant class variability [13, 34]. Ideally, this threshold value will allow for only a small false discovery rate in class differences, typically $10-20\%$ [7, 13, 41].

One issue that has arisen in class comparison in cancer is that the natural variation in cancer often leads to important entities being overlooked, because they are notably different from normals only in a subset of the cancers under study. This has led to the development of outlier profile analysis, which can identify entities that differ only in a subset of cancer samples when compared to normals [59, 164].

Once the entities that distinguish the classes are identified, it is often desirable to link these factors to the specific biological processes that are responsible for inducing the class differences. Often, these algorithms identify sets of genes differentially expressed in the classes. The biological function of these genes can then be established by comparing to lists of these functions encoded in *Gene Ontology* using various algorithms (e.g., hypergeometric tests) or prepackaged software (e.g., Onto-Express [39]). Enrichment relative to random assortment in the genes in a set can indicate the biological mechanisms driving the class distinction, although it is important to remember that the processes may only be correlative. For instance, in cancer studies, it is not unusual to find enhancement in cell cycle processes. This is a reflection of the fact that cancer cells are actively growing, but it does not provide insight into the mechanisms driving the development of cancer.

25.3.4.2 Clustering

Clustering algorithms can be used in cancer systems biology to identify sets of genes or samples representative of tumor types. When applied to group rows of **X**, clustering algorithms seek groups of biological entities (e.g., genes) that behave similarly across the columns (samples). Clustering algorithms can also group the columns of **X** to identify the samples that are similar (often tumor or tissue type). While the former analysis is used to identify genes that behave similarly and may be linked through biological processes active in some samples, the latter identifies genes whose changes may be indicative of the type of sample (e.g., cancer subtype). Regardless of the application, there are two main classes of clustering algorithms: *unsupervised* algorithms, which use the measurements to define and assign groups and *supervised* algorithms, which assign genes or samples to predefined groups based on additional information [7, 132].

Several unsupervised algorithms, including hierarchical clustering, k-means clustering, and self-organizing maps (SOM) [123, 132], commonly infer sets of entities that co-vary within a population. Among the most common algorithms is agglomerative hierarchical clustering algorithms, which iteratively group pairs or clusters that are closest by a user-defined metric [43]. Relative similarity is often represented graphically in a dendrogram, and clusters result from choosing a distance along the dendrogram to divide the entities into groups. If a fixed number of clusters is expected, K-means or K-medians clustering can be used. These algorithms first randomly assign measurements from each entity into a cluster and then iteratively reshuffle the members of the cluster to maximize the inter-cluster distance between the mean or median value of cluster members [157, 158]. Self-organizing maps (SOM) are another method to create clusters, and here additional information can be gained by the relationship between clusters [161]. Unsupervised clustering algorithms have been extended for microarray data to group genes involved in inhibitory or activating interactions to implicate regulatory relationships [34].

Supervised clustering (also known as classification or class prediction) is used to identify groups of entities that differentiate specific classes. These algorithms are often employed in similar scenarios to class distinction algorithms to identify the patterns that relate to the observed classes. Whereas class distinction consider biological entities individually, supervised clustering algorithms infer these patterns by simultaneously considering groups of biological entities and their interactions [13]. Typically, supervised algorithms will rely on data divided into a training set and a test set. The algorithm first finds the entities and their relationships to distinguish the classes, and then uses the test set to validate the results. Many supervised clustering techniques are adopted from standard algorithms [123, 132], including notably artificial neural networks [88] and support vector machines (SVM) [27].

Clustering genes can aid in identifying potential clinical multigene biomarkers. In addition, ontological categories in genes grouped together may provide insight into biological processes driving the disease, effectively identifying specific subprocesses related to the hallmarks of cancer. Gene set enrichment [159] and other techniques can look for enhancement of ontological categories in clusters. Clustering in samples can identify subtypes of cancer that could correlate with phenotypic responses beyond standard cancer staging, potentially providing insights that could improve cancer treatment.

Clustering algorithms will link entities or samples that behave similarly throughout an entire experiment. However, it is sometimes the case that a group of genes will be coordinated in transcript levels over only part of an experiment, or that a series of samples share behavior only across some genes. Biclustering approaches attempt to address this by identifying subclusters that link subsets of entities and samples [162]. In this case, comparing inferred subclusters of genes may implicate molecular differences between individual samples that could be responsible for different therapeutic responses in the measured population, perhaps due to different drivers of the cancer hallmarks.

25.3.4.3 Pattern Recognition

One weakness in using clustering for analysis of biological data is that entities must belong to a single cluster. Therefore, these algorithms often cannot represent the biological reality that molecules such as genes and proteins often have multiple uses, requiring their assignment to multiple groups when addressing function [118]. To address this, we differentiate clustering (gene in one class) from pattern recognition (gene in multiple classes) here, although these terms are not well separated in general use. Since genes are reused in multiple processes, pattern recognition algorithms may be more adept than clustering algorithms at inferring the biological processes in cancer. Nonetheless, clustering algorithms remain more adept for biomarker discovery, which requires finding entities that tie strongly to class.

These pattern recognition algorithms can be viewed as performing matrix decomposition on $\mathbf{X}$, as in

$$\mathbf{X} = \mathbf{AP}, \tag{25.1}$$

where the rows of $\mathbf{P}$ form a set of basis vectors that represent the underlying biological behaviors and the columns of $\mathbf{A}$ provide the corresponding assignment of genes to the behaviors. One standard algorithm for such matrix decomposition is principal component analysis (PCA) [8] . In spite of the wide applicability of PCA for many scientific applications, the orthogonality requirement on its basis vectors prevents PCA from representing the overlapping patterns that arise from reuse of genes in biological processes [118].

Three algorithms were developed simultaneously to address this problem in the domains of medical spectral imaging, Bayesian Decomposition (BD) [124], computer vision, Non-negative Matrix Factorization (NMF) [101], and statistics, Bayesian Factor Regression Modeling (BFRM) [31]. BD and NMF algorithms rely on positivity in the **A** and **P** matrices, and both have been applied to microarray data analysis [91, 118]. BFRM extends the model of Eq. 25.1 by allowing additional terms related to phenotype [31]. From a statistical point of view, BD and BFRM have an advantage in that the Markov chain Monte Carlo procedure underlying them is less prone to becoming trapped at local maxima in the probability space. In addition, they allow incorporation of biological knowledge in the decomposition, such as known pathways or transcriptional coregulation in BD [96] or phenotypic measurements in BFRM.

The complexity of gene regulation, with many if not all genes having multiple transcription factors regulating their expression, plays an especially important role in cellular responses to cell signaling changes. The complex signaling networks (e.g., Fig. 25.3) drive overlapping sets of transcriptional regulators, which in turn regulate overlapping sets of genes. Isolating the transcriptional signature of one factor requires solving Eq. 25.1, so that the correct set of genes are grouped together. These then become surrogates for estimation of transcription factor activity, which can be integrated with other measurements (e.g., proteomics, receptor status) to improve estimations of signaling changes. Such signaling changes, often arising from mutations or epigenetics, play a key role in driving most of the cancer hallmarks.

25.4 Discussion

Cancer results from a series of events, usually mutations or epigenetic modifications, that lead to undesirable gain or loss of protein activity. In each individual, the driving mutations of the cancer differ slightly. Nonetheless, these mutations are linked through their effects on the hallmarks described in Sect. 25.2.1. The interaction of the biological processes driving cancer development and growth is complex, and often involves feedback between the heterogeneous components of a tumor and its environment, such as hormonal changes in the individual. As a result, the processes responsible for an individual's cancer are often difficult to identify, and are even more challenging to target therapeutically, requiring a systems approach to understanding this disease.

The field of systems biology offers opportunities for quantitatively testing and formulating hypotheses about the underlying system-wide processes occurring in tumorigenesis. To date, most effort has focused on developing algorithms to identify differences between tumor subtypes and normal tissues on the basis of a single type of measurement in order to implicate the biological processes that are active in cancer. However, due to the complexity of the system, greater insight will be gleaned from algorithms that can infer biological processes from multiple measurements of different molecular species.

Another approach, predictive computational models, provides an additional source of experiments to integrate, test, and propose biological theories about the cancer system. Traditional computational models rely on biochemical models based on the master chemical equation in its stochastic form suitable for low concentrations [61] and its successors [105]. More novel models rely on graphical models of networks of interacting proteins or networks of phenomenological responses (i.e., upregulation of a gene), with Bayesian Networks being the most widely used [139]. Such models provide a simulation of the system, potentially allowing *in silico* testing of hypotheses, which should streamline development of therapy. However, these models presently are too incomplete to handle the complexity of the full cancer system.

To fully utilize the measurements of the cancer system, systems biology must formulate data integration algorithms that incorporate data from multiple technologies and prior information from biological studies in predictive mathematical models. This combination of information will provide the algorithm with the statistical power to infer biological activity in tumorigenesis in spite of the complexity underlying the entire cancer system. Integrated algorithms could quantitatively identify the driving mutations or aberrant processes in an individual cancer and, therefore, reach the fundamental goal of personalized medicine by suggesting an individualized treatment plan.

References

1. Ach, R. A., Wang, H., & Curry, B. (2008). Measuring micrornas: Comparisons of microarray and quantitative PCR measurements, and of different total RNA prep methods. *BMC Biotechnology*, *8*, 69.
2. Ahmed, F., Steele, J. C., Herbert, J. M. J., Steven, N. M., & Bicknell, R. (2008, September). Tumor stroma as a target in cancer. *Current Cancer Drug Targets*, *8*(6), 447–453.
3. Al-Hajj, M., Wicha, M. S., Benito-Hernandez, A., Morrison, S. J., & Clarke, M. F. (2003, April 1). Prospective identification of tumorigenic breast cancer cells. *Proceedings of the National Academy of Sciences of the United States of America*, *100*(7), 3983–3988.
4. Alberts, B. (2008). *Molecular biology of the cell* (5th ed.). New York: Garland Science.
5. Albini, A., & Sporn, M. B. (2007, February). The tumour microenvironment as a target for chemoprevention. *Nature Reviews. Cancer*, *7*(2), 139–147.
6. Ali, S., & Ali, S. (2007, October 15). Role of c-KIT/SCF in cause and treatment of gastrointestinal stromal tumors (GIST). *Gene*, *401*(1–2), 38–45.
7. Allison, D. B., Cui, X., Page, G. P., & Sabripour, M. (2006, January). Microarray data analysis: From disarray to consolidation and consensus. *Nature Reviews. Genetics*, *7*(1), 55–65.
8. Alter, O., Brown, P. O., & Botstein, D. (2000, August 29). Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences of the United States of America*, *97*(18), 10101–10106.
9. Anderson, A. R. A., & Quaranta, V. (2008, March). Integrative mathematical oncology. *Nature Reviews. Cancer*, *8*(3), 227–234.

10. Anderson, A. R. A., Weaver, A. M., Cummings, P. T., & Quaranta, V. (2006, December 1). Tumor morphology and phenotypic evolution driven by selective pressure from the microenvironment. *Cell*, *127*(5), 905–915.
11. Bacac, M., Provero, P., Mayran, N., Stehle, J.-C., Fusco, C., & Stamenkovic, I. (2006). A mouse stromal response to tumor invasion predicts prostate and breast cancer patient survival. *PLoS ONE*, *1*, e32.
12. Baggerly, K. A., Morris, J. S., Edmonson, S. R., & Coombes, K. R. (2005, February). Signal in noise: Evaluating reported reproducibility of serum proteomic tests for ovarian cancer. *Journal of the National Cancer Institute*, *97*(4), 307–309.
13. Baker, S. G.,& Kramer, B. S. (2008, October). Using microarrays to study the microenvironment in tumor biology: The crucial role of statistics. *Seminars in Cancer Biology*, *18*(5), 305–310.
14. Barrett, T., Troup, D. B., Wilhite, S. E., Ledoux, P., Rudnev, D., Evangelista, C., Kim, I. F., Soboleva, A., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Muertter, R. N., & Edgar, R. (2009, January). NCBI GEO: Archive for high-throughput functional genomic data. *Nucleic Acids Research*, *37*(Database issue), D885–D890.
15. Bauch, A., & Superti-Furga, G. (2006, April). Charting protein complexes, signaling pathways, and networks in the immune system. *Immunological Reviews*, *210*, 187–207.
16. Baylin, S. B., & Ohm, J. E. (2006, February). Epigenetic gene silencing in cancer – a mechanism for early oncogenic pathway addiction? *Nature Reviews. Cancer*, *6*(2), 107–116.
17. Beerenwinkel, N., Antal, T., Dingli, D., Traulsen, A., Kinzler, K. W., Velculescu, V. E., Vogelstein, B., & Nowak, M. A. (2007). Genetic progression and the waiting time to cancer. *PLoS Computational Biology*, *3*(11), e225.
18. Behmoaram, E., Bijian, K., Bismar, T. A., & Alaoui-Jamali, M. A. (2008). Early stage cancer cell invasion: Signaling, biomarkers and therapeutic targeting. *Frontiers in Bioscience*, *13*, 6314–6325.
19. Bernard, A., & Hartemink, A. J. (2005). Informative structure priors: Joint learning of dynamic regulatory networks from multiple types of data. *Pacific Symposium on Biocomputing*, *10*, 459–470.
20. Bibikova, M., Lin, Z., Zhou, L., Chudin, E., Garcia, E. W., Wu, B., Doucet, D., Thomas, N. J., Wang, Y., Vollmer, E., Goldmann, T., Seifart, C., Jiang, W., Barker, D. L., Chee, M. S., Floros, J., & Fan, J.-B. (2006, March). High-throughput DNA methylation profiling using universal bead arrays. *Genome Research*, *16*(3), 383–393.
21. Bidaut, G., Suhre, K., Claverie, J.-M., & Ochs, M. F. (2006). Determination of strongly overlapping signaling activity from microarray data. *BMC Bioinformatics*, *7*, 99.
22. Bleyer, A., Barr, R., Hayes-Lattin, B., Thomas, D., Ellis, C., & Anderson, B. (2008, April). The distinctive biology of cancer in adolescents and young adults. *Nature Reviews. Cancer*, *8*(4), 288–298.
23. Blow, J. J., & Gillespie, P. J. (2008, October). Replication licensing and cancer–a fatal entanglement? *Nature Reviews. Cancer*, *8*(10), 799–806.
24. Bock, C., & Lengauer, T. (2008, January 1). Computational epigenetics. *Bioinformatics*, *24*(1), 1–10.
25. Bodenreider, O., & Stevens, R. (2006, September). Bio-ontologies: Current trends and future directions. *Brief Bioinformatics*, *7*(3), 256–274.
26. Brazhnik, P., de la Fuente, A., & Mendes, P. (2002). Gene networks: How to put the function in genomics. *Trends Biotechnology*, *20*(11), 467–472.
27. Brown, M. P., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares, M., Jr., & Haussler, D. (2000, January 4). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America*, *97*(1), 262–267.
28. Cai, L., Dalal, C. K., & Elowitz, M. B. (2008, September 25). Frequency-modulated nuclear localization bursts coordinate gene regulation. *Nature*, *455*(7212), 485–490.
29. Cairns, R. A., Khokha, R., & Hill, R. P. (2003, November). Molecular mechanisms of tumor invasion and metastasis: An integrated view. *Current Molecular Medicine*, *3*(7), 659–671.

30. Camacho, D., Vera Licona, P., Mendes, P., & Laubenbacher, R. (2007, December). Comparison of reverse-engineering methods using an in silico network. *Annals of the New York Academy of Sciences*, *1115*, 73–89, .
31. Carvalho, C. M., Chang, J., Lucas, J., Nevins, J. R., Wang, Q., & West, M. (2008). High-dimensional sparse factor modelling: Applications in gene expression genomics. *Journal of the American Statistical Association*, *103*, 1438–1456.
32. Chiang, A. C., & Massague, J. (2008, December 25). Molecular basis of metastasis. *The New England Journal of Medicine*, *359*(26), 2814–2823.
33. Clasper, S., Royston, D., Baban, D., Cao, Y., Ewers, S., Butz, S., Vestweber, D., & Jackson, D. G. (2008, September 15). A novel gene expression profile in lymphatics associated with tumor growth and nodal metastasis. *Cancer Research*, *68*(18), 7293–7303.
34. Claverie, J. M. (1999). Computational methods for the identification of differential and coordinated gene expression. *Human Molecular Genetics*, *8*(10), 1821–1832.
35. Croce, C. M. (2008, January 31). Oncogenes and cancer. *The New England Journal of Medicine*, *358*(5), 502–511.
36. Crosby, M. E., Kulshreshtha, R., Ivan, M., & Glazer, P. M. (2009, February 1). MicroRNA regulation of DNA repair gene expression in hypoxic stress. *Cancer Research*, *69*(3), 1221–1229.
37. De Wever, O., Demetter, P., Mareel, M., & Bracke, M. (2008, November 15). Stromal myofibroblasts are drivers of invasive cancer growth. *International Journal of Cancer*, *123*(10), 2229–2238.
38. Dick, J. E. (2008, December 15). Stem cell concepts renew cancer research. *Blood*, *112*(13), 4793–4807.
39. Draghici, S., Khatri, P., Bhavsar, P., Shah, A., Krawetz, S. A., & Tainsky, M. A. (2003, July). Onto-tools, the toolkit of the modern biologist: Onto-express, onto-compare, onto-design and onto-translate. *Nucleic Acids Research*, *31*(13), 3775–3781.
40. Duesberg, P., & Li, R. (2003, May-June). Multistep carcinogenesis: A chain reaction of aneuploidizations. *Cell Cycle*, *2*(3), 202–210.
41. Dupuy, A., & Simon, R. M. (2007, January 17). Critical review of published microarray studies for cancer outcome and guidelines on statistical analysis and reporting. *Journal of the National Cancer Institute*, *99*(2), 147–157.
42. Dvorak, H. F. (1986, December 25). Tumors: Wounds that do not heal. similarities between tumor stroma generation and wound healing. *The New England Journal of Medicine*, *315*(26), 1650–1659.
43. Eisen, M. B., Spellman, P. T., Brown, P. O., & Botstein, D. (1998, December 8). Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, *95*(25), 14863–14868.
44. Eklund, A. C., & Szallasi, Z. (2008). Correction of technical bias in clinical microarray data improves concordance with known biological information. *Genome Biology*, *9*(2), R26.
45. Eyler, C. E., & Rich, J. N. (2008, June 10). Survival of the fittest: Cancer stem cells in therapeutic resistance and angiogenesis. *Journal of Clinical Oncology*, *26*(17), 2839–2845.
46. Fearon, E. R., & Vogelstein, B. (1990, June 1). A genetic model for colorectal tumorigenesis. *Cell*, *61*(5), 759–767.
47. Feinberg, A. P., Ohlsson, R., & Henikoff, S. (2006, January). The epigenetic progenitor origin of human cancer. *Nature Reviews. Genetics*, *7*(1), 21–33.
48. Feinberg, A. P., & Tycko, B. (2004, February). The history of cancer epigenetics. *Nature Reviews. Cancer*, *4*(2), 143–153.
49. Finak, G., Bertos, N., Pepin, F., Sadekova, S., Souleimanova, M., Zhao, H., Chen, H., Omeroglu, G., Meterissian, S., Omeroglu, A., Hallett, M., & Park, M. (2008, May). Stromal gene expression predicts clinical outcome in breast cancer. *Nature Medicine*, *14*(5), 518–527.
50. Finkel, T., Serrano, M., & Blasco, M. A. (2007, August 16). The common biology of cancer and ageing. *Nature*, *448*(7155), 767–774.
51. Folkman, J. (2006). Angiogenesis. *Annual Review of Medicine*, *57*, 1–18.

52. Frisk, T., Foukakis, T., Dwight, T., Lundberg, J., Hoog, A., Wallin, G., Eng, C., Zedenius, J., & Larsson, C. (2002, September). Silencing of the pten tumor-suppressor gene in anaplastic thyroid cancer. *Genes Chromosomes Cancer*, *35*(1), 74–80.
53. Gao, P., Honkela, A., Rattray, M., & Lawrence, N. D. (2008, August 15). Gaussian process modelling of latent chemical species: Applications to inferring transcription factor activities. *Bioinformatics*, *24*(16), i70–i75.
54. Gargiulo, G., & Minucci, S. (2009). Epigenomic profiling of cancer cells. *International Journal of Biochemistry and Cell Biology*, *41*(1), 127–135.
55. Gatenby, R. A., & Vincent, T. L. (2003, October 1). An evolutionary model of carcinogenesis. *Cancer Research*, *63*(19), 6212–6220.
56. Ge, H., Walhout, A. J. M., & Vidal, M. (2003, October). Integrating 'omic' information: A bridge between genomics and systems biology. *Trends in Genetics*, *19*(10), 551–560.
57. Gentleman, R. C., Carey, V. J., Bates, D. M., Bolstad, B., Dettling, M., Dudoit, S., Ellis, B., Gautier, L., Ge, Y., Gentry, J., Hornik, K., Hothorn, T., Huber, W., Iacus, S., Irizarry, R., Leisch, F., Li, C., Maechler, M., Rossini, A. J., Sawitzki, G., Smith, C., Smyth, G., Tierney, L., Yang, J. Y., & Zhang, J. (2004). Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, *5*(10), R80.
58. Gevertz, J. L., & Torquato, S., (2006, December 21). Modeling the effects of vasculature evolution on early brain tumor growth. *Journal of Theoretical Biology*, *243*(4), 517–531.
59. Ghosh, D., & Chinnaiyan, A. M. (2009, January). Genomic outlier profile analysis: Mixture models, null hypotheses, and nonparametric estimation. *Biostatistics*, *10*(1), 60–69.
60. Giehl, K., & Menke, A. (2008). Microenvironmental regulation of E-Cadherin-mediated adherens junctions. *Frontiers in Bioscience*, *13*, 3975–3985.
61. Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, *22*, 403–434.
62. Gius, D., & Spitz, D. R. (2006, July-August). Redox signaling in cancer biology. *Antioxid Redox Signal*, *8*(7–8), 1249–1252.
63. Greenman, C., Stephens, P., Smith, R., Dalgliesh, G. L., Hunter, C., Bignell, G., Davies, H., Teague, J., Butler, A., Stevens, C., Edkins, S., O'Meara, S., Vastrik, I., Schmidt, E. E., Avis, T., Barthorpe, S., Bhamra, G., Buck, G., Choudhury, B., Clements, J., Cole, J., Dicks, E., Forbes, S., Gray, K., Halliday, K., Harrison, R., Hills, K., Hinton, J., Jenkinson, A., Jones, D., Menzies, A., Mironenko, T., Perry, J., Raine, K., Richardson, D., Shepherd, R., Small, A., Tofts, C., Varian, J., Webb, T., West, S., Widaa, S., Yates, A., Cahill, D. P., Louis, D. N., Goldstraw, P., Nicholson, A. G., Brasseur, F., Looijenga, L., Weber, B. L., Chiew, Y.-E., DeFazio, A., Greaves, M. F., Green, A. R., Campbell, P., Birney, E., Easton, D. F., Chenevix-Trench, G., Tan, M.-H., Khoo, S. K., Teh, B. T., Yuen, S. T., Leung, S. Y., Wooster, R., Futreal, P. A., & Stratton, M. R. (2007, March 8). Patterns of somatic mutation in human cancer genomes. *Nature*, *446*(7132), 153–158.
64. Griffin, T. J., Gygi, S. P., Ideker, T., Rist, B., Eng, J., Hood, L., & Aebersold, R. (2002, April). Complementary profiling of gene expression at the transcriptome and proteome levels in Saccharomyces cerevisiae. *Molecular & Cellular Proteomics*, *1*(4), 323–333.
65. Hahn, W. C., & Weinberg, R. A. (2002, November 14). Rules for making human tumor cells. *The New England Journal of Medicine*, *347*(20), 1593–1603.
66. Hanahan, D., & Weinberg, R. A. (2000). The hallmarks of cancer. *Cell*, *100*(1), 57–70.
67. Hanawalt, P. C., & Spivak, G. (2008, December). Transcription-coupled DNA repair: Two decades of progress and surprises. *Nature Reviews. Molecular Cell Biology*, *9*(12), 958–970.
68. Harley, C. B. (2008, March). Telomerase and cancer therapeutics. *Nature Reviews. Cancer*, *8*(3), 167–179.
69. Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., & Young, R. A. (2001). Maximum likelihood estimate of optimal scaling factors for expression array normalization. *SPIE BiOS*. Proc SPIE *4266*, 132–140.
70. Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., & Young, R. A. (2001). Using graphical models and genomic expression data to statistically validate models of genetic regulatory networks. *Pacific Symposium on Biocomputing*, *6*, 422–433.

71. Herbert, J. M. J., Stekel, D., Sanderson, S., Heath, V. L., & Bicknell, R. (2008). A novel method of differential gene expression analysis using multiple cdna libraries applied to the identification of tumour endothelial genes. *BMC Genomics*, *9*, 153.
72. Herman, J. G., & Baylin, S. B. (2003, November 20). Gene silencing in cancer in association with promoter hypermethylation. *The New England Journal of Medicine*, *349*(21), 2042–2054.
73. Hornberg, J. J., Bruggeman, F. J., Westerhoff, H. V., & Lankelma, J. (2006, February-March). Cancer: A systems biology disease. *Biosystems*, *83*(2–3), 81–90.
74. Hua, F., Hautaniemi, S., Yokoo, R., & Lauffenburger, D. A. (2006, August 22). Integrated mechanistic and data-driven modelling for multivariate analysis of signalling pathways. *Journal of The Royal Society Interface*, *3*(9), 515–526.
75. Hunter, D. J., Kraft, P., Jacobs, K. B., Cox, D. G., Yeager, M., Hankinson, S. E., Wacholder, S., Wang, Z., Welch, R., Hutchinson, A., Wang, J., Yu, K., Chatterjee, N., Orr, N., Willett, W. C., Colditz, G. A., Ziegler, R. G., Berg, C. D., Buys, S. S., McCarty, C. A., Feigelson, H. S., Calle, E. E., Thun, M. J., Hayes, R. B., Tucker, M., Gerhard, D. S., Fraumeni, J. F., Jr., Hoover, R. N., Thomas, G., & Chanock, S. J. (2007, July). A genome-wide association study identifies alleles in FGFR2 associated with risk of sporadic postmenopausal breast cancer. *Nature Genetics*, *39*(7), 870–874.
76. Hwang, D., Rust, A. G., Ramsey, S., Smith, J. J., Leslie, D. M., Weston, A. D., de Atauri, P., Aitchison, J. D., Hood, L., Siegel, A. F., & Bolouri, H. (2005, November 29). A data integration methodology for systems biology. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(48), 17296–17301.
77. Iacobuzio-Donahue, C. A., Ryu, B., Hruban, R. H., & Kern, S. E. (2002, January). Exploring the host desmoplastic response to pancreatic carcinoma: Gene expression of stromal and neoplastic cells at the site of primary invasion. *The American Journal of Pathology*, *160*(1), 91–99.
78. Irizarry, R. A., Hobbs, B., Collin, F., Beazer-Barclay, Y. D., Antonellis, K. J., Scherf, U., & Speed, T. P. (2003, April). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, *4*(2), 249–264.
79. Irizarry, R. A., Ladd-Acosta, C., Wen, B., Wu, Z., Montano, C., Onyango, P., Cui, H., Gabo, K., Rongione, M., Webster, M., Ji, H., Potash, J. B., Sabunciyan, S., & Feinberg, A. P. (2009, February). The human colon cancer methylome shows similar hypo- and hypermethylation at conserved tissue-specific cpg island shores. *Nature Genetics*, *41*(2), 178–186.
80. Irmer, D., Funk, J. O., & Blaukat, A. (2007, August 23). EGFR kinase domain mutations – functional impact and relevance for lung cancer therapy. *Oncogene*, *26*(39), 5693–5701.
81. Jacks, T., & Weinberg, R. A. (2002, December 27). Taking the study of cancer cell survival to a new dimension. *Cell*, *111*(7), 923–925.
82. Jemal, A., Siegel, R., Ward, E., Hao, Y., Xu, J., Murray, T., & Thun, M. J. (2008, March-April). Cancer statistics. *CA: A Cancer Journal for Clinicians*, *58*(2), 71–96.
83. Johnson, W. E., Li, C., & Rabinovic, A. (2007, January). Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, *8*(1), 118–127.
84. Jones, S., Zhang, X., Parsons, D. W., Lin, J. C.-H., Leary, R. J., Angenendt, P., Mankoo, P., Carter, H., Kamiyama, H., Jimeno, A., Hong, S.-M., Fu, B., Lin, M.-T., Calhoun, E. S., Kamiyama, M., Walter, K., Nikolskaya, T., Nikolsky, Y., Hartigan, J., Smith, D. R., Hidalgo, M., Leach, S. D., Klein, A. P., Jaffee, E. M., Goggins, M., Maitra, A., Iacobuzio-Donahue, C., Eshleman, J. R., Kern, S. E., Hruban, R. H., Karchin, R., Papadopoulos, N., Parmigiani, G., Vogelstein, B., Velculescu, V. E., & Kinzler, K. W. (2008, September 26). Core signaling pathways in human pancreatic cancers revealed by global genomic analyses. *Science*, *321*(5897), 1801–1806.
85. Jordan, J. D., Landau, E. M., & Iyengar, R. (2000, October). Signaling networks: The origins of cellular multitasking. *Cell*, *103*, 193–200.
86. Keniry, M., & Parsons, R. (2008, September 18). The role of PTEN signaling perturbations in cancer and in targeted therapy. *Oncogene*, *27*(41), 5477–5485.
87. Khalil, I. G., & Hill, C. (2005, January). Systems biology for cancer. *Current Opinion in Oncology*, *17*(1), 44–48.

88. Khan, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C. R., Peterson, C., & Meltzer, P. S. (2001, June). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, *7*(6), 673–679.
89. Kholodenko, B. N., Kiyatkin, A., Bruggeman, F. J., Sontag, E., Westerhoff, H. V., & Hoek, J. B. (2002, October 1). Untangling the wires: A strategy to trace functional interactions in signaling and gene networks. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(20), 12841–12846.
90. Kiberstis, P. A., & Travis, J. (2006). Celebrating a glass half-full. *Science*, *312*, 1157.
91. Kim, P. M., & Tidor, B. (2003, July). Subsystem identification through dimensionality reduction of large-scale gene expression data. *Genome Research*, *13*(7), 1706–1718.
92. Kinzler, K. W., & Vogelstein, B. (1996, October 18). Lessons from hereditary colorectal cancer. *Cell*, *87*(2), 159–170.
93. Knudson, A. G. (1993, December 1). Antioncogenes and human cancer. *Proceedings of the National Academy of Sciences of the United States of America*, *90*(23), 10914–10921.
94. Kolch, W. (2000). Meaningful relationships: The regulation of the Ras/Raf/MEK/ERK pathway by protein interactions. *The Biochemical Journal*, *351*(Pt. 2), 289–305.
95. Komarova, N. L., & Wodarz, D. (2004, May 4). The optimal rate of chromosome loss for the inactivation of tumor suppressor genes in cancer. *Proceedings of the National Academy of Sciences of the United States of America*, *101*(18), 7017–7021.
96. Kossenkov, A. V., Peterson, A. J., & Ochs, M. F. (2007). Determining transcription factor activity from microarray data using Bayesian Markov chain Monte Carlo sampling. *Studies in Health Technology and Informatics*, *129*(Pt. 2), 1250–1254.
97. Kundu, J. K., & Surh, Y.-J. (2008, July-August). Inflammation: Gearing the journey to cancer. *Mutation Research*, *659*(1–2), 15–30.
98. Lagasse, E. (2008, January). Cancer stem cells with genetic instability: The best vehicle with the best engine for cancer. *Gene Therapy*, *15*(2), 136–142.
99. Lahav, G., Rosenfeld, N., Sigal, A., Geva-Zatorsky, N., Levine, A. J., Elowitz, M. B., & Alon, U. (2004, February). Dynamics of the p53-MDM2 feedback loop in individual cells. *Nature Genetics*, *36*(2), 147–150.
100. Lamy, P., Andersen, C. L., Wikman, F. P., & Wiuf, C. (2006). Genotyping and annotation of Affymetrix SNP arrays. *Nucleic Acids Research*, *34*(14), e100.
101. Lee, D. D., & Seung, H. S. (1999, October). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*(6755), 788–791.
102. Leek, J. T., & Storey, J. D. (2007, September). Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genetics*, *3*(9), 1724–1735.
103. Lewis, M. T. (2008, October). Faith, heresy and the cancer stem cell hypothesis. *Future Oncology*, *4*(5), 585–589.
104. Li, H., Fan, X., & Houghton, J. (2007, July 1). Tumor microenvironment: The role of the tumor stroma in cancer. *Journal of Cellular Biochemistry*, *101*(4), 805–815.
105. Li, H., Cao, Y., Petzold, L. R., & Gillespie, D. T. (2008). Algorithms and software for stochastic simulation of biochemical reacting systems. *Biotechnology Progress*, *24*(1), 56–61.
106. Lister, R., O'Malley, R. C., Tonti-Filippini, J., Gregory, B. D., Berry, C. C., Millar, A. H., & Ecker, J. R. (2008, May 2). Highly integrated single-base resolution maps of the epigenome in arabidopsis. *Cell*, *133*(3), 523–536.
107. Liu, E. T., Kuznetsov, V. A., & Miller, L. D. (2006, April). In the pursuit of complexity: Systems medicine in cancer biology. *Cancer Cell*, *9*(4), 245–247.
108. Liu, H. G., You, J., Pan, Y. F., Hu, X. Q., Huang, D. P., & Zhang, X. H. (2009, January 7). Cancer stem cell hierarchy. *Stem Cell Reviews*. *5*, 174.
109. Lockhart, D. J., Dong, H., Byrne, M. C., Follettie, M. T., Gallo, M. V., Chee, M. S., Mittmann, M., Wang, C., Kobayashi, M., Horton, H., & Brown, E. L. (1996, December). Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, *14*(13), 1675–1680.
110. Loeb, L. A., Bielas, J. H., & Beckman, R. A. (2008, May 15). Cancers exhibit a mutator phenotype: Clinical implications. *Cancer Research*, *68*(10), 3551–3557.

111. Macdonald, F., Ford, C. H. J., & Casson, A. G. (2004). *Molecular biology of cancer* (2nd ed.). London: BIOS Scientific Publishers.
112. Mantovani, A. (2009, January 1). Cancer: Inflaming metastasis. *Nature*, *457*(7225), 36–37.
113. Markevich, N. I., Moehren, G., Demin, O. V., Kiyatkin, A., Hoek, J. B., & Kholodenko, B. N. (2004, June). Signal processing at the Ras circuit: What shapes Ras activation patterns? *Systems Biology, IEEE Proceedings*, *1*(1), 104–113.
114. Michor, F. (2008). Mathematical models of cancer stem cells. *Journal of Clinical Oncology*, *26*, 2854–2861.
115. Michor, F., Iwasa, Y., Vogelstein, B., Lengauer, C., & Nowak, M. A. (2005, Feburary). Can chromosomal instability initiate tumorigenesis? *Seminars in Cancer Biology*, *15*(1), 43–49.
116. Mikkelsen, T. S., Hanna, J., Zhang, X., Ku, M., Wernig, M., Schorderet, P., Bernstein, B. E., Jaenisch, R., Lander, E. S., & Meissner, A. (2008, July 3). Dissecting direct reprogramming through integrative genomic analysis. *Nature*, *454*(7200), 49–55.
117. Mirmohammadsadegh, A., Marini, A., Nambiar, S., Hassan, M., Tannapfel, A., Ruzicka, T., & Hengge, U. R. (2006, July 1). Epigenetic silencing of the PTEN gene in melanoma. *Cancer Research*, *66*(13), 6546–6552.
118. Moloshok, T. D., Klevecz, R. R., Grant, J. D., Manion, F. J., Speier, W. F., IV., & Ochs, M. F. (2002). Application of Bayesian decomposition for analysing microarray data. *Bioinformatics*, *18*(4), 566–575.
119. Morange, M. (2007, December 6). The field of cancer research: An indicator of present transformations in biology. *Oncogene*, *26*(55), 7607–7610.
120. Mukherjee, S., & Speed, T. P. (2008, September 23). Network inference using informative priors. *Proceedings of the National Academy of Sciences of the United States of America*, *105*(38), 14313–14318.
121. Nielsen, T. O., West, R. B., Linn, S. C., Alter, O., Knowling, M. A., O'Connell, J. X., Zhu, S., Fero, M., Sherlock, G., Pollack, J. R., Brown, P. O., Botstein, D., & van de Rijn, M. (2002, April 13). Molecular characterisation of soft tissue tumours: A gene expression study. *Lancet*, *359*(9314), 1301–1307.
122. Norton, L., & Massague, J. (2006, August). Is cancer a disease of self-seeding? *Nature Medicine*, *12*(8), 875–878.
123. Ochs, M. F., & Godwin, A. K. (2003, March). Microarrays in cancer: Research and applications. *Biotechniques*, (Suppl.), *34*, S4–S15.
124. Ochs, M. F., Stoyanova, R. S., Arias-Mendoza, F., & Brown, T. R. (1999). A new method for spectral decomposition using a bilinear Bayesian approach. *Journal of Magnetic Resonance*, *137*(1), 161–176.
125. Ochs, M. F., Rink, L., Tarn, C., Mburu, S., Taguchi, T., Eisenberg, B., & Godwin, A. K. (2009). Detection of treatment-induced changes in signaling pathways in gastrointestinal stromal tumors using transcriptomic data. *Cancer Research*, *69*(23). 9125–9132.
126. Overall, C. M., & Lopez-Otin, C. (2002, September). Strategies for MMP inhibition in cancer: Innovations for the post-trial era. *Nature Reviews. Cancer*, *2*(9), 657–672.
127. Paget, S. (1889). The distribution of secondary growths in cancer of the breast. *Lancet*, *1*, 571–573.
128. Papin, J. A., Hunter, T., Palsson, B. O., & Subramaniam, S. (2005, February). Reconstruction of cellular signalling networks and analysis of their properties. *Nature Reviews. Molecular Cell Biology*, *6*(2), 99–111.
129. Parkinson, H., Kapushesky, M., Kolesnikov, N., Rustici, G., Shojatalab, M., Abeygunawardena, N., Berube, H., Dylag, M., Emam, I., Farne, A., Holloway, E., Lukk, M., Malone, J., Mani, R., Pilicheva, E., Rayner, T. F., Rezwan, F., Sharma, A., Williams, E., Bradley, X. Z., Adamusiak, T., Brandizi, M., Burdett, T., Coulson, R., Krestyaninova, M., Kurnosov, P., Maguire, E., Neogi, S. G., Rocca-Serra, P., Sansone, S.-A., Sklyar, N., Zhao, M., Sarkans, U., & Brazma, A. (2009, January). ArrayExpress update–from an archive of functional genomics experiments to the atlas of gene expression. *Nucleic Acids Research*, *37*(Database issue), D868–D872.

130. Parsons, D. W., Jones, S., Zhang, X., Lin, J. C.-H., Leary, R. J., Angenendt, P., Mankoo, P., Carter, H., Siu, I.-M., Gallia, G. L., Olivi, A., McLendon, R., Rasheed, B. A., Keir, S., Nikolskaya, T., Nikolsky, Y., Busam, D. A., Tekleab, H., Diaz, L. A., Jr., Hartigan, J., Smith, D. R., Strausberg, R. L., Marie, S. K. N., Shinjo, S. M. O., Yan, H., Riggins, G. J., Bigner, D. D., Karchin, R., Papadopoulos, N., Parmigiani, G., Vogelstein, B., Velculescu, V. E., & Kinzler, K. W. (2008, September 26). An integrated genomic analysis of human glioblastoma multiforme. *Science*, *321*(5897), 1807–1812.
131. Podhajcer, O. L., Benedetti, L. G., Girotti, M. R., Prada, F., Salvatierra, E., & Llera, A. S. (2008, December). The role of the matricellular protein SPARC in the dynamic interaction between the tumor and the host. *Cancer Metastasis Reviews*, *27*(4), 691–705.
132. Quackenbush, J. (2001, June). Computational analysis of microarray data. *Nature Reviews. Genetics*, *2*(6), 418–427.
133. Rajagopalan, H., Nowak, M. A., Vogelstein, B., & Lengauer, C. (2003, September). The significance of unstable chromosomes in colorectal cancer. *Nature Reviews. Cancer*, *3*(9), 695–701.
134. Reya, T., Morrison, S. J., Clarke, M. F., & Weissman, I. L. (2001, November 1). Stem cells, cancer, and cancer stem cells. *Nature*, *414*(6859), 105–111.
135. Ribba, B., Saut, O., Colin, T., Bresch, D., Grenier, E., & Boissel, J. P. (2006, December 21). A multiscale mathematical model of avascular tumor growth to investigate the therapeutic benefit of anti-invasive agents. *Journal of Theoretical Biology*, *243*(4), 532–541.
136. Ricke, R. M., van Ree, J. H., & van Deursen, J. M. (2008, September). Whole chromosome instability and cancer: A complex relationship. *Trends in Genetics*, *24*(9), 457–466.
137. Roberts, C. J., Nelson, B., Marton, M. J., Stoughton, R., Meyer, M. R., Bennett, H. A., He, Y. D., Dai, H., Walker, W. L., Hughes, T. R., Tyers, M., Boone, C., & Friend, S. H. (2000, February 4). Signaling and circuitry of multiple MAPK pathways revealed by a matrix of global gene expression profiles. *Science*, *287*(5454), 873–880.
138. Ross, D. T., Scherf, U., Eisen, M. B., Perou, C. M., Rees, C., Spellman, P., Iyer, V., Jeffrey, S. S., Van de Rijn, M., Waltham, M., Pergamenschikov, A., Lee, J. C., Lashkari, D., Shalon, D., Myers, T. G., Weinstein, J. N., Botstein, D., & Brown, P. O. (2000, March). Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics*, *24*(3), 227–235.
139. Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., & Nolan, G. P. (2005). Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, *308*(5721), 523–529.
140. Sahai, E. (2007, October). Illuminating the metastatic process. *Nature Reviews. Cancer*, *7*(10), 737–749.
141. Sandhu, J. S. (2008, July). Prostate cancer and chronic prostatitis. *Current Urology Reports*, *9*(4), 328–332.
142. Santos, S. D. M., Verveer, P. J., & Bastiaens, P. I. H. (2007, March). Growth factor-induced MAPK network topology shapes ERK response determining PC-12 cell fate. *Nature Cell Biology*, *9*(3), 324–330.
143. Saul, Z. M., & Filkov, V. (2007). Exploring biological network structure using exponential random graph models. *Bioinformatics*, *23*(19), 2604–2611.
144. Sauro, H. M., & Kholodenko, B. N. (2004). Quantitative analysis of signaling networks. *Progress in Biophysics and Molecular Biology*, *86*, 5–43.
145. Schadt, E. E., Li, C., Ellis, B., & Wong, W. H. (2001). Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data. *Journal of Cellular Biochemistry. Supplement*, (Suppl. 37), 120–125.
146. Schena, M., Shalon, D., Davis, R. W., & Brown, P. O. (1995, October 20). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, *270*(5235), 467–470.
147. Schena, M., Shalon, D., Heller, R., Chai, A., Brown, P. O., & Davis, R. W. (1996, October 1). Parallel human genome analysis: Microarray-based expression monitoring of 1,000 genes. *Proceedings of the National Academy of Sciences of the United States of America*, *93*(20), 10614–10619.

148. Schrader, A. J., Lechner, O., Templin, M., Dittmar, K. E. J., Machtens, S., Mengel, M., Probst-Kepper, M., Franzke, A., Wollensak, T., Gatzlaff, P., Atzpodien, J., Buer, J., & Lauber, J. (2002, April 22). CXCR4/CXCL12 expression and signalling in kidney cancer. *British Journal of Cancer*, *86*(8), 1250–1256.
149. Segal, E., Yelensky, R., & Koller, D. (2003). Genome-wide discovery of transcriptional modules from DNA sequence and gene expression. *Bioinformatics*, *19*(Suppl. 1), i273–i282.
150. Sell, S. (2007, January). Cancer and stem cell signaling: A guide to preventive and therapeutic strategies for cancer stem cells. *Stem Cell Reviews*, *3*(1), 1–6.
151. Sharma, S. V., Bell, D. W., Settleman, J., & Haber, D. A. (2007, March). Epidermal growth factor receptor mutations in lung cancer. *Nature Reviews. Cancer*, *7*(3), 169–181.
152. Shipitsin, M., & Polyak, K. (2008, May). The cancer stem cell hypothesis: In search of definitions, markers, and relevance. *Laboratory Investigation*, *88*(5), 459–463.
153. Simpson, R. J., & Dorow, D. S. (2001). Cancer proteomics: From signaling networks to tumor markers. *Trends in Biotechnology*, *19*(Suppl. 10), S40–S48.
154. Singh, S. K., Clarke, I. D., Hide, T., & Dirks, P. B. (2004, September 20). Cancer stem cells in nervous system tumors. *Oncogene*, *23*(43), 7267–7273.
155. Smith, I. M., Glazer, C. A., Mithani, S. K., Ochs, M. F., Sun, W., Bhan, S., Vostrov, A., Abdullaev, Z., Lobanenkov, V., Gray, A., Liu, C., Chang, S. S., Ostrow, K. L., Westra, W. H., Begum, S., Dhara, M., & Califano, J. (2009). Coordinated activation of candidate proto-oncogenes and cancer testes antigens via promoter demethylation in head and neck cancer and lung cancer. *PLoS ONE*, *4*(3), e4961.
156. Soria, J.-C., Lee, H.-Y., Lee, J. I., Wang, L., Issa, J.-P., Kemp, B. L., Liu, D. D., Kurie, J. M., Mao, L., & Khuri, F. R. (2002, May). Lack of PTEN expression in non-small cell lung cancer could be related to promoter methylation. *Clinical Cancer Research*, *8*(5), 1178–1184.
157. Soukas, A., Cohen, P., Socci, N. D., & Friedman, J. M. (2000, April 15). Leptin-specific patterns of gene expression in white adipose tissue. *Genes & Development*, *14*(8), 963–980.
158. Soukas, A., Socci, N. D., Saatkamp, B. D., Novelli, S., & Friedman, J. M. (2001, September 7). Distinct transcriptional profiles of adipogenesis in vivo and in vitro. *Journal of Chemical Biology*, *276*(36), 34167–34174.
159. Subramanian, A., Tamayo, P., Mootha, V. K., Mukherjee, S., Ebert, B. L., Gillette, M. A., Paulovich, A., Pomeroy, S. L., Golub, T. R., Lander, E. S., & Mesirov, J. P. (2005, October). Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(43), 15545–15550.
160. Tamada, Y., Kim, S., Bannai, H., Imoto, S., Tashiro, K., Kuhara, S., & Miyano, S. (2003, October). Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. *Bioinformatics*, *19*(Suppl. 2), ii227–ii236.
161. Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. S., & Golub, T. R. (1999, March 16). Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Sciences of the United States of America*, *96*(6), 2907–2912.
162. Tanay, A., Sharan, R., & Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, *18*(Suppl. 1), S136–S144.
163. The Cancer Genome Atlas Research Network. (2008, October 23). Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, *455*(7216), 1061–1068.
164. Tibshirani, R., & Hastie, T. (2007). Outlier sums for differential gene expression analysis. *Biostatistics*, *8*(1), 2–8.
165. Tozeren, A., Coward, C. W., & Petushi, S. P. (2005, March 7). Origins and evolution of cell phenotypes in breast tumors. *Journal of Theoretical Biology*, *233*(1), 43–54.
166. van Steensel, B. (2005, June). Mapping of genetic and epigenetic regulatory networks using microarrays. *Nature Genetics*, *37*(Suppl.), S18–S24.
167. van't Veer, L. J., & Bernards, R. (2008). Enabling personalized cancer medicine through analysis of gene-expression patterns. *Nature*, *452*(7187), 564–570.

168. Varambally, S., Yu, J., Laxman, B., Rhodes, D. R., Mehra, R., Tomlins, S. A., Shah, R. B., Chandran, U., Monzon, F. A., Becich, M. J., Wei, J. T., Pienta, K. J., Ghosh, D., Rubin, M. A., & Chinnaiyan, A. M. (2005, November). Integrative genomic and proteomic analysis of prostate cancer reveals signatures of metastatic progression. *Cancer Cell*, *8*(5), 393–406.
169. Vazquez, A., Bond, E. E., Levine, A. J., & Bond, G. L. (2008, December). The genetics of the p53 pathway, apoptosis and cancer therapy. *Nature Reviews. Drug Discovery*, *7*(12), 979–987.
170. Ventura, A. C., Jackson, T. L., & Merajver, S. D. (2009, January 15). On the role of cell signaling models in cancer research. *Cancer Research*, *69*(2), 400–402.
171. Vezzoni, L., & Parmiani, G. (2008, September). Limitations of the cancer stem cell theory. *Cytotechnology*, *58*(1), 3–9.
172. Visvader, J. E., & Lindeman, G. J. (2008, October). Cancer stem cells in solid tumours: Accumulating evidence and unresolved questions. *Nature Reviews. Cancer*, *8*(10), 755–768.
173. Vogelstein, B., & Kinzler, K. W. (2004, August). Cancer genes and the pathways they control. *Nature Medicine*, *10*(8), 789–799.
174. von Mering, C., Jensen, L. J., Kuhn, M., Chaffron, S., Doerks, T., Kruger, B., Snel, B., & Bork, P. (2007). String 7–recent developments in the integration and prediction of protein interactions. *Nucleic Acids Research*, *35*(Database issue), D358–D362.
175. Weaver, B. A. A., & Cleveland, D. W. (2006, December). Does aneuploidy cause cancer? *Current Opinion in Cell Biology*, *18*(6), 658–667.
176. Weinberg, R. A. (2008, June). Mechanisms of malignant progression. *Carcinogenesis*, *29*(6), 1092–1095.
177. Werbowetski-Ogilvie, T. E., & Bhatia, M. (2008, August). Pluripotent human stem cell lines: What we can learn about cancer initiation. *Trends in Molecular Medicine*, *14*(8), 323–332.
178. Werhli, A. V., & Husmeier, D. (2007). Reconstructing gene regulatory networks with Bayesian networks by combining expression data with multiple sources of prior knowledge. *Statistical Applications in Genetics and Molecular Biology*, *6*, Article15.
179. Xing, Y., Kapur, K., & Wong, W. H. (2006). Probe selection and expression index computation of Affymetrix exon arrays. *PLoS ONE*, *1*, e88.
180. Zakharkin, S. O., Kim, K., Mehta, T., Chen, L., Barnes, S., Scheirer, K. E., Parrish, R. S., Allison, D. B., & Page, G. P. (2005). Sources of variation in Affymetrix microarray experiments. *BMC Bioinformatics*, *6*, 214.
181. Zhao, H., Ramos, C. F., Brooks, J. D., & Peehl, D. M. (2007, January). Distinctive gene expression of prostatic stromal cells cultured from diseased versus normal tissues. *Journal of Cellular Physiology*, *210*(1), 111–121.
182. Zhou, X. J., Kao, M.-C. J., Huang, H., Wong, A., Nunez-Iglesias, J., Primig, M., Aparicio, O. M., Finch, C. E., Morgan, T. E., & Wong, W. H. (2005, February). Functional annotation and network reconstruction through cross-platform integration of microarray data. *Nature Biotechnology*, *23*(2), 238–243.

Chapter 26
Comparative Genomics

Xuhua Xia

Abstract Comparative genomics was previously misguided by the naïve dogma that what is true in *E. coli* is also true in the elephant. With the rejection of such a dogma, comparative genomics has been positioned in proper evolutionary context. Here I numerically illustrate the application of phylogeny-based comparative methods in comparative genomics involving both continuous and discrete characters to solve problems from characterizing functional association of genes to detection of horizontal gene transfer and viral genome recombination, together with a detailed explanation and numerical illustration of statistical significance tests based on the false discovery rate (FDR). FDR methods are essential for multiple comparisons associated with almost any large-scale comparative genomic studies. I discuss the strength and weakness of the methods and provide some guidelines on their proper applications.

26.1 Introduction

The development of comparative genomics predates the availability of genomic sequences. It has long been known that organisms are related, with many homologous genes sharing similar functions among diverse organisms. For example, the yeast *IRA2* gene is homologous to the human *NF1* gene, and the functional equivalence of the two genes was demonstrated by the yeast *IRA2* mutant being rescued by the human *NF1* gene [5]. This suggests the possibility that simple genomes can be used as a model to study complicated genomes. A multitude of such demonstrations of functional equivalence of homologous genes across diverse organisms has led to the dogmatic assertion that what is true in *E. coli* is also true in the elephant [attributed to Jacques Monod, [33], p. 290].

X. Xia
Department of Biology, University of Ottawa, Ottawa, Canada
e-mail: Xuhua.Xia@uottawa.ca

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_26,

© Springer-Verlag Berlin Heidelberg 2011

It is the realization that what is true in *E. coli* is often not true in the elephant that has brought comparative genomics into the proper evolutionary context. The impact of this realization on comparative genomics is best illustrated by a simple example. Suppose we compare a Cadillac Deville and a Dodge Caravan. The two are similar in functionality except that the Caddy warns the driver when it is backing towards an object behind the car. What is the structural basis of this warning function? Nearly all structural elements in the Caddy have their 'homologues' in the Dodge Caravan except for the four sensors on the rear bumper. This would lead us to quickly hypothesize that the four sensors are associated with the warning function, which turns out to be true. Now if we replace the Dodge Caravan with a baby stroller, then the comparison will be quite difficult because a stroller and a Caddy differ structurally in numerous ways and any structural difference could be responsible for the warning function. We may mistakenly hypothesize that the rear lights, the antenna or the rear window defroster in the Caddy, which are all missing in the stroller, may be responsible for the warning function. To test the hypotheses, we would destroy the rear lights, the antenna, the rear window defroster, etc., one by one, but will get nothing but negative results. What could be even worse is that, when destroying the rear lights, we accidentally destroy a part of the electric system in such a way that the warning function is lost, which would mislead us to conclude that the rear lights are indeed part of the structural basis responsible for the warning function-an 'experimentally substantiated' yet wrong conclusion. A claim that what is true in *E. coli* is also true in the elephant is equivalent to a claim that what is true in the stroller is also true in the Caddy. It will take comparative genomics out of its proper conceptual framework in evolutionary biology.

Evolutionary theory states that all genetic variation, including genomic variation, results from two sculptors of nature, i.e., mutation (including recombination) and selection. Thus, any genomic difference can be attributed to differences in differential mutation and selection pressure. This allows us not only to characterize evolutionary changes along different evolutionary lineages, but also to seek evolutionary processes underlying the character changes. In particular, evolutionary biology provides the proper comparative methods [7,20,28,55,71] for comparative genomics.

In what follows, I will numerically illustrate the comparative methods for analyzing genomic features that are either continuous or discrete. Large-scale comparative genomic studies almost always lead to multiple comparisons. So I will also illustrate the computation involved in controlling for false discovery rate which represents a key development in recent studies of statistical significance tests [8,9]. One evolutionary process that has shaped bacterial genomes is the horizontal gene transfer, and the phylogenetic incongruence test used to detect such horizontal gene transfer events will be illustrated. The last section covers comparative genomic methods for detecting recombination events and mapping recombination points.

While molecular phylogenetics is often essential in comparative genomics, the subject has been treated fully elsewhere [22, 50, 66]. Simple overviews of the subject are also available [4, 66, 87]. A more egregious omission in this chapter is genome rearrangement, but interested readers may consult the publications of my colleague at University of Ottawa, David Sankoff, who is a pioneer in the field and wrote excellent reviews on the subject [69, 70]. A large-scale empirical study of genome rearrangement in yeast species following a whole-genome duplication (WGD) event, featuring a meticulous reconstruction of gene order of the ancestral genome before WGD, has recently been published [26].

26.2 The comparative Method for Continuous Characters

26.2.1 Variation in Genomic GC% Among Bacterial Species

Studies of the variation in genomic GC% among bacterial species serve as the easiest entry point into comparative genomics. Wide variation in genomic GC% is observed in bacterial species. A popular selectionist hypothesis is that bacterial species living in high temperature should have high genomic GC% for two reasons. First, an increased GC usage, with more hydrogen bonds between the two DNA strands, would stabilize the physical structure of the genome [42, 64]. Second, high temperature would need more thermostable amino acids [3] which are typically coded by GC-rich codons. This implies that genomic GC% should increase with optimal grow temperature (OGT) in bacterial species. While this prediction is not supported, either based on results of sequence analysis [24] or by experimental studies [94], it has been found that GC% of rRNA genes is highly correlated with OGT [24, 30, 49, 79], [18, p. 535]. In particular, when the loop and stem regions of rRNA are studied separately, it was found that the hyperthermophilic bacterial species not only have higher proportion of GC in the stems but also longer stems [80]. In contrast, the GC% in the loop region correlates only weakly with OGT. Because stems function to stabilize the RNA secondary structure which is functionally important, these results are consistent with the hypothesized selection for RNA structural stability in high environmental temperatures.

When studying the relationship between two quantitative variables, such as OGT and stem GC%, a phylogeny-based comparison is crucially important to avoid violation of statistical assumptions. Figure 26.1 illustrates a case in which one may mistakenly conclude a positive relationship between X and Y when the 16 data points are taken as independent. A phylogenetic tree superimposed on the points allows us to see immediately that the data points are not independent. All eight points in the left share one common ancestor, so do the eight points in the right. So the superficial association between X and Y could be due to a single coincidental change in X and Y in one of the two common ancestors. One needs to use the phylogeny-based method, such as independent contrasts [20], [22, pp. 432–459] or the generalized least-squares method [46, 56, 57] when assessing the relationship between quantitative variables.

While the derivation and mathematical justification of the phylogeny-based comparative method is quite complicated, the most fundamental assumption is the

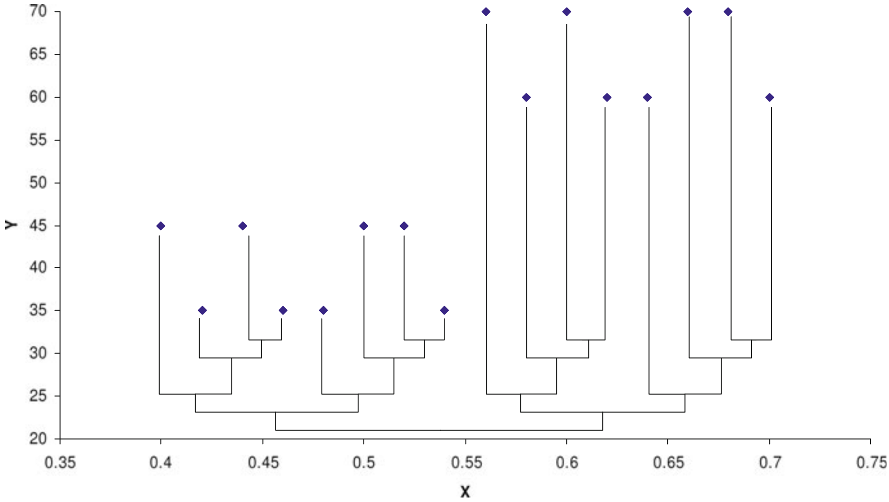


Fig. 26.1 Phylogeny-based comparison is important for evolutionary studies. The data points, when wrongly taken as independent, would result in a significant positive but spurious relationship between Y and X (which represent any two continuous variables, e.g., GC% and OGT)

Brownian motion model [22, pp. 391–414] which appears reasonable for neutrally evolving continuous characters. Here I illustrate the actual computation of independent contrasts with a numerical example to facilitate its application to comparative genomics, prompted by my personal belief that one generally cannot interpret the results properly if one does not know how the results are obtained.

Suppose a phylogeny of eight bacterial species whose OGT and GC% of rRNA genes have been measured, with the eight species referred to hereafter as s_1 to s_8 from left to right in Fig. 26.2. The computation is recursive, and is exactly the same for any quantitative variable. So we will only illustrate the computation involving OGT. One may repeat the computation involving GC% as an exercise.

The computation is of three steps. First, we recursively compute the ancestral values for internal (ancestral) nodes x_1 to x_6. We treat these ancestors as if they were new taxa and compute the branch lengths leading to these ancestral nodes. We may start with the two sister species s_1 and s_2. The OGT of their ancestor (x_1) is a weighted average of the OGT values for s_1 and s_2 (weighted by the branch lengths):

$$OGT_{x_1} = \frac{v_2}{v_1 + v_2} OGT_{s_1} + \frac{v_1}{v_1 + v_2} OGT_{s_2} = \frac{3 \times 70}{4} + \frac{1 \times 74}{4} = 71 \quad (26.1)$$

One may note that the weighting scheme in (26.1) is such that the ancestral state is more similar to the state of the descendent node with a shorter branch than the other with a longer branch. This makes intuitive sense as a descendent node diverged much from the ancestor should be less reliable for inferring the ancestral state than a descendent node diverged little from the ancestor.

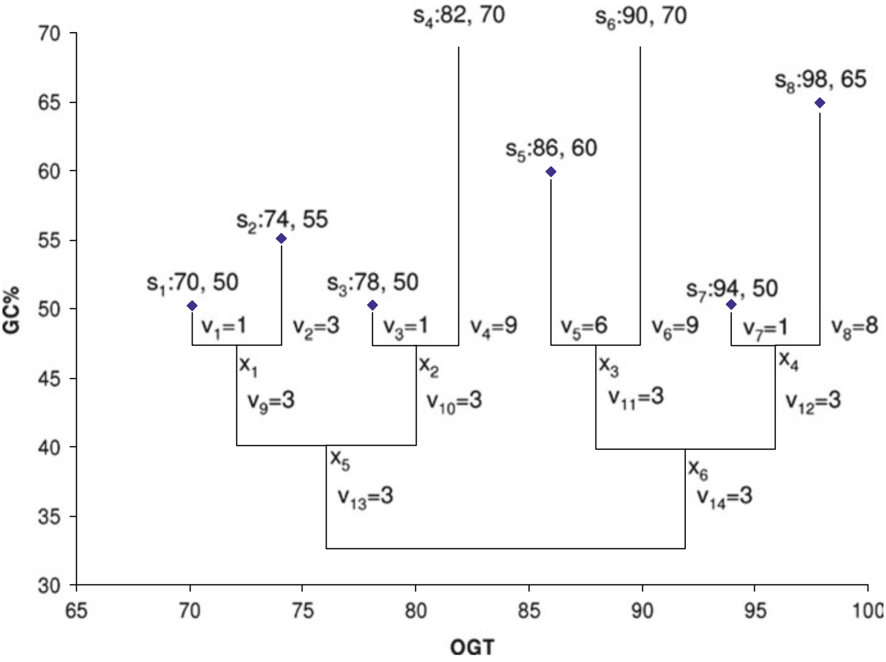


Fig. 26.2 A phylogeny of eight bacterial species (s_1–s_8) each labeled with optimal growth temperature (OGT) and GC% of the stem region of rRNA genes in the format of 'OGT, GC%'. The branch lengths ($v_1 - v_{14}$) are next to the branches. Ancestral nodes are designated by x_1 to x_6

We now treat x_1 as if it is a new taxon and compute the branch lengths leading to it from its ancestor (x_5) as

$$v_{x_1} = \frac{v_1 v_2}{v_1 + v_2} + v_9 = \frac{1 \times 3}{1 + 3} + 3 = 3.75 \tag{26.2}$$

We do the same for x_2 to x_4, and the associated OGT_{xi} and v_{xi} values are listed in Table 26.1. The computation of the ancestral states for x_5 and x_6 is similar to that in (26.1), e.g.,

$$OGT_{x_5} = \frac{v_{x_2} OGT_{x_1}}{v_{x_1} + v_{x_2}} + \frac{v_{x_1} OGT_{x_2}}{v_{x_1} + v_{x_2}} = \frac{3.9 \times 71}{7.65} + \frac{3.75 \times 78.4}{7.65} = 74.63 \tag{26.3}$$

Now we can take the second step to compute the unweighted contrasts (designated by C) as well as the sum of branch lengths linking the two contrasted taxa. With eight species, we have seven (= $n - 1$, where n is the number of species) contrasts (first column in Table 26.2). These unweighted contrasts, as well as the sum of branch lengths (SumV) associated with the contrasts, are illustrated for those between s_1 and s_2 and between x_1 and x_2 for OGT in (26.4). All the computed unweighted contrasts for both OGT and GC%, as well as the associated SumV

Table 26.1 Computed ancestral states (OGT_{xi} and GC_{xi}) and the branch lengths (v_{xi}) for the six ancestral nodes

x_i	OGT_{x_i}	v_{x_i}	GC_{x_i}
x_1	71.0000	3.7500	51.2500
x_2	78.4000	3.9000	52.0000
x_3	87.6000	6.6000	64.0000
x_4	94.4444	3.8889	51.6667
x_5	74.6275	4.9118	51.6176
x_6	91.9068	5.4470	56.2394

Table 26.2 Unweighted and weight contrasts for the two quantitative variables OGT and GC%

Contrast	Unweighted Contrasts		SumV	Weighted Contrasts	
	OGT	GC%		WC_{OGT}	$WC_{GC\%}$
$s_1 - s_2$	−4.0000	−5.0000	4.0000	−2.0000	−2.5000
$s_3 - s_4$	−4.0000	−20.0000	10.0000	−1.2649	−6.3246
$s_5 - s_6$	−4.0000	−10.0000	15.0000	−1.0328	−2.5820
$s_7 - s_8$	−4.0000	−15.0000	9.0000	−1.3333	−5.0000
$x_1 - x_2$	−7.4000	−0.7500	7.6500	−2.6755	−0.2712
$x_3 - x_4$	−6.8444	12.3333	10.4889	−2.1134	3.8082
$x_5 - x_6$	−17.279	4.6218	10.3588	−5.3687	−1.4360

values, are listed in columns 2–4 in Table 26.2.

$$\begin{aligned} C_{s_1-s_2 OGT} &= OGT_{s_1} - OGT_{s_2} = 70 - 74 = -4 \\ SumV_{C_{s_1-s_2}} &= v_1 + v_2 = 1 + 3 = 4 \\ C_{x_1-x_2 OGT} &= OGT_{x_1} - OGT_{x_2} = 71 - 78.4 = -7.4 \\ SumV_{C_{x_1-x_2}} &= v_{x_1} + v_{x_2} = 3.75 + 3.9 = 7.65 \end{aligned} \tag{26.4}$$

We can now take the third step of obtaining independent weighted contrasts (WC) by dividing each unweighted contrasts by the square root of the associated SumV. For example,

$$\begin{aligned} WC_{s_1-s_2 OGT} &= \frac{C_{s_1-s_2 OGT}}{\sqrt{Sum\ V_{s_1-s_2}}} = \frac{-4}{\sqrt{4}} = -2 \\ WC_{x_1-x_2 OGT} &= \frac{C_{x_1-x_2 OGT}}{\sqrt{Sum\ V_{x_1-x_2}}} = \frac{-7.4}{\sqrt{7.65}} = -2.6755 \end{aligned} \tag{26.5}$$

These independent contrasts for OGT thus computed, together with those for GC%, are shown in the last two columns in Table 26.2. Now we need to assess the relationship between WC_{OGT} and $WC_{GC\%}$, specifically whether an increase in OGT will result in an increase in GC%, i.e., whether the two are positively correlated. There are two ways to assess the relationship. The first is parametric by performing a linear regression of $WC_{GC\%}$ on WC_{OGT}, forcing the intercept equal to 0. The reason for a zero intercept is that we do not expect a change in GC% if there is no change in OGT. The resulting slope is 0.4647. The regression accounts for 11.17% of the variation in $WC_{GC\%}$. The square root of 11.17%, equal to 0.3342, is the correlation coefficient between the two. Of course you may also do a regression of WC_{OGT} on $WC_{GC\%}$, which will result in a slope of 0.2403. These slopes and the correlation coefficients are in the default output in the CONTRAST program in PHYLIP [21]. The relationship between WC_{OGT} and $WC_{GC\%}$, although positive, is not significant ($p = 0.4249$).

One may also assess the relationship between WC_{OGT} and $WC_{GC\%}$ by using nonparametric tests. For example, we expect half of the (WC_{OGT}, $WC_{GC\%}$) pairs to have the same sign (i.e., both positive or both negative) and the other half to have different signs. We observe six pairs to have the same sign and one pair to have different signs (Table 26.2). So we have

$$\chi^2 = \frac{(6-3.5)^2}{3.5} + \frac{(1-3.5)^2}{3.5} = 3.5714 \tag{26.6}$$

With one degree of freedom, the relationship is not significant ($p = 0.05878$).

Although the method of independent contrasts has been available for many years, many studies, even recent ones, still fall into the same trap, as illustrated in Fig. 26.1, of concluding a significant relationship between X and Y without taking the phylogeny into account. A recent claim of a strong relationship between intron conservation and intron number [32] represents one of such studies.

One shortcoming of the method of independent contrasts is that the value of the ancestral state is always somewhere between the two values of the descendents. This implies that it cannot detect directional changes over time. For example, if the ancestor is small in body size and all descendents have increased in body size over time, then the Brownian motion model assumed by the independent contrast method is no longer applicable. In such cases, one should use the generalized least square method [46, 56, 57].

When the method of independent contrasts was applied to the real data to assess the relationship between bacterial OGT and GC% of rRNA stem sequences and between OGT and rRNA stem lengths, the two relationships are both statistically significant [80]. Thus, the selectionist hypothesis is supported, but it accounts for only a very small fraction of variation in the genomic GC% among bacterial species, which calls for an alternative hypothesis for the variation in genomic GC%.

The mutation hypothesis of genomic GC% variation [48, 76, 94, 96] invokes biased mutation in different bacterial species to explain genomic variation in GC%, i.e., GC-rich genomes are the result of GC-biased mutation. One prediction from the mutation hypothesis is that the third codon position should increase more rapidly with the genomic GC% than the first codon position which in turn should have its GC% increase more rapidly with the genomic GC% than the second codon position. The reason for this prediction is that the third codon positions are little constrained functionally because most substitutions at the third codon positions are synonymous. Some nucleotide substitutions at the first codon positions are synonymous, but most are nonsynonymous. All nucleotide substitutions at the second

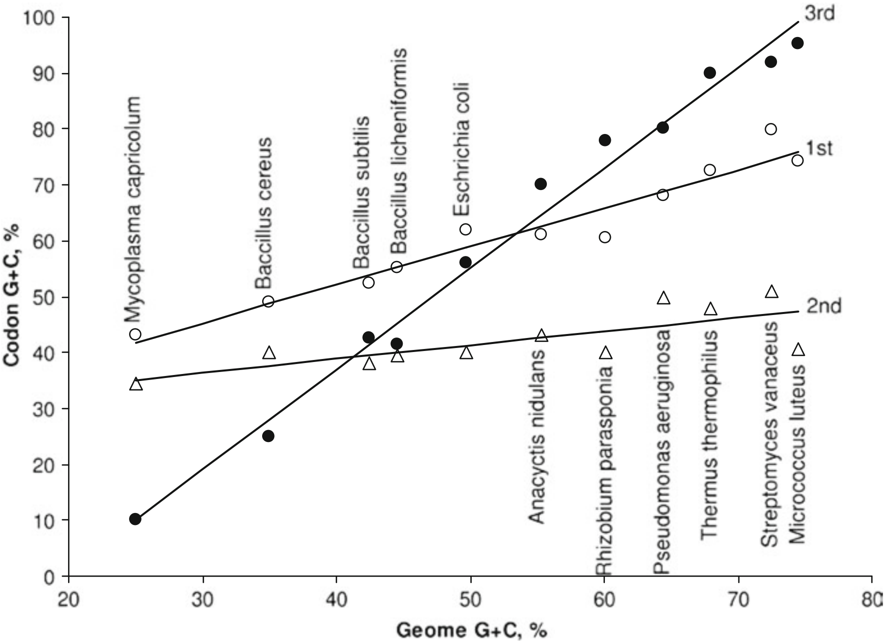


Fig. 26.3 Correlation of GC% between genomic DNA and first, second and third codon positions [48]. While the actual position of the points may be substantially revised with new genomic data (e.g., the GC% for the first, second and third codon positions for *Mycoplasma capricolum* is 35.8%, 27.4%, and 8.8% based on all annotated CDSs in the genomic sequence), the general trend remains the same

codon positions are nonsynonymous and typically involve rather different amino acids [83,91]. The empirical results [48] strongly support this prediction (Fig. 26.3).

The pattern in Fig. 26.3, while consistent with the mutation hypothesis, has resulted in two misconceptions. First, the pattern shown by the third codon position is often interpreted to reflect mutation bias. This interpretation is incorrect because the third codon position is subject to selection by differential availability of tRNA species [16, 82, 86, 88, 90]. We may contrast a GC-rich *Streptomyces coelicolor* and a GC-poor *Mycoplasma capricolum* as an illustrative example. *M. capricolum* has no tRNA with a C or G at the wobble site for four-fold codon families (Ala, Gly, Pro, Thr and Val), i.e., the translation machinery would be inefficient in translating C-ending or G-ending codons. This implies selection in favour of A-ending or U-ending codons and will consequently reduce GC% at the third codon position. This most likely has contributed to the low GC% at the third codon position in *M. capricolum*. In contrast, most of the tRNA genes translating the five four-fold codon families in the GC-rich *S. coelicolor* have G or C at the wobble site, and should favour the use of C-ending or G-ending codons. This most likely has contributed to the high GC% at the third codon position in *S. coelicolor*. The same pattern is observed for two-fold codon families. The most conspicuous one is the Gln codon family (CAA and CAG). There is only one *tRNA*^{Gln} gene in *M. capricolum* with a UUG anticodon favouring the CAA codon. In contrast, there are two $tRNA^{Gln}$ in *S. coelicolor*, both with a CUG anticodon favouring the CAG codon. Thus, the high slope for the third codon position in Fig. 26.3 is at least partially attributable to the tRNA-mediated selection. Relative contribution of mutation and tRNA-mediated selection to codon usage has been evaluated in several recent studies [16, 86, 88, 90].

Second, the observation that GC% of the third codon position increases with genomic GC% is sometimes taken to imply that the frequency of G-ending and C-ending codons will increase with genomic GC% or GC-biased mutation [40]. This is not generally true. Take the arginine codons for example. Given the transition probability matrix for the six synonymous codons shown in Table 26.3, the equilibrium frequencies (π) for the six codons are

$$\begin{aligned}
\pi_{AGA} &= \frac{1}{2k^2 + 3k + 1} \\
\pi_{AGG} = \pi_{CGA} = \pi_{CGT} &= \frac{k}{2k^2 + 3k + 1} \\
\pi_{CGC} = \pi_{CGG} &= \frac{k^2}{2k^2 + 3k + 1}
\end{aligned} \tag{26.7}$$

The three solutions correspond to the number of GC in the codon, with AGA having one, AGG, CGA and CGT having two, and CGC and CGG having three G or C. One may note that the G-ending codon AGG has the same equilibrium frequency as that of the A-ending CGA and the T-ending CGT. Thus, we should not expect A-ending or T-ending codons to always decrease, or G-ending and C-ending codons always increase, with increasing genomic GC% or GC-biased mutation. In fact, according to the solutions in (26.7), AGG, CGA, and CGT will first increase with k until k reaches $\sqrt{2}/2$, and will then decrease with k when $k > \sqrt{2}/2$.

Table 26.3 Transition probability matrix for the six synonymous arginine codons, with α for transitions (C↔ T and A ↔ G), β for transversions, and k modeling AT-biased mutation ($0 \leq k \leq 1$) or GC-biased mutation ($k > 1$). We ignore nonsynonymous substitutions because nonsynonymous substitution rate is often negligibly low compared to synonymous rate. The diagonal is constrained by the row sum equal to 1

	CGT	CGC	CGA	CGG	AGA	AGG
CGT		$k\alpha$	β	$k\beta$	0	0
CGC	α		β	β	0	0
CGA	β	$k\beta$		$k\alpha$	β	0
CGG	β	β	α		0	β
AGA	0	0	$k\beta$	0		$k\alpha$
AGG	0	0	0	$k\beta$	α	

One may ask why the phylogeny-based comparison was not used for characterizing the relationship between codon GC% and genomic GC% in the 11 species in Fig. 26.3. The reason is that the two variables change very fast relative to the divergence time among the studied species, i.e., phylogenetic relatedness among the 11 species is a poor predictor of the codon GC% or genomic GC%. That genomic GC% has little phylogenetic inertia is generally true in prokaryotic species [93]. In such cases, one may assume approximate data independence and perform a phylogeny-free analysis. Another study that leads to insight into the relationship between UV exposure and GC% in bacterial genomes [73], which may be the first comparative genomic study, is also not phylogeny-based.

26.3 DNA Methylation, CpG Dinucleotide Frequencies and GC Content

CpG deficiency has been documented in a large number of genomes covering a wide taxonomic distribution [15, 35–37, 53]. DNA methylation is one of the many hypotheses proposed to explain differential CpG deficiency in different genomes [10, 62, 77]. It features a plausible mechanism as follows. Methyltransferases in many species, especially those in vertebrates, appear to methylate specifically the cytosine in CpG dinucleotides, and the methylated cytosine is prone to mutate to thymine by spontaneous deamination [23, 44]. This implies that CpG would gradually decay into TpG and CpA, leading to CpG deficiency and reduced genomic GC%. Different genomes may differ in CpG deficiency because they differ in methylation activities, with genomes having high methylation activities exhibiting stronger CpG deficiency than genomes with little or no methylation activity.

In spite of its plausibility, the methylation-deamination hypothesis has several major empirical difficulties (e.g., [15]), especially in recent years with genome-based analysis (e.g., Goto et al. 2000). For example, *Mycoplasma genitalium* does not seem to have any methyltransferase and exhibits no methylation activity, yet its genome shows a severe CpG deficiency. Therefore, the CpG deficiency in *M. genitalium*, according to the critics of the methylation-deamination hypothesis, must be due to factors other than DNA methylation.

A related species, *M. pneumoniae*, also devoid of any DNA methyltransferase, has a genome that is not deficient in CpG. Given the difference in CpG deficiency between the two Mycoplasma species, the methylation hypothesis would have predicted that the *M. genitalium* genome is more methylated than the *M. pneumoniae* genome, which is not true as neither has a methyltransferase. Thus, the methylation hypothesis does not seem to have any explanatory power to account for the variation in CpG deficiency, at least in the Mycoplasma species.

These criticisms are derived from phylogeny-free reasoning. When phylogeny-based comparisons are made, the Mycoplasma genomes become quite consistent with the methylation hypothesis [85]. First, several lines of evidence suggest that the common ancestor of *M. genitalium* and *M. pneumoniae* have methyltransferases

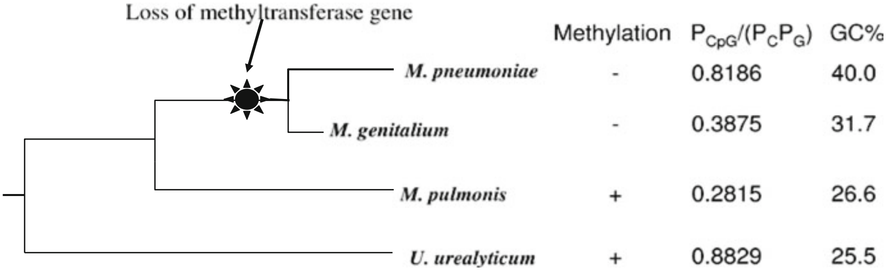


Fig. 26.4 Phylogenetic tree of Mycoplasma pneumoniae, *M. genitaliums*, and their relatives, together with the presence (+) or absence (−) of CpG-specific methylation, $P_{CpG}/(P_C P_G)$ as a measure of CpG deficiency, and genomic GC%. *M. pneumoniae* evolves faster and has a longer branch than *M. genitalium*

methylating C in CpG dinucleotides, and should have evolved strong CpG deficiency and low genomic GC% as a result of the specific DNA methylation. Methylated m^5C exists in the DNA of a close relative, *Mycoplasma hyorhinis* [61], suggesting the existence of methyltransferases in *M. hyorhinis*. Methyltransferases are present in *Mycoplasma pulmonis* which contains at least four CpG-specific methyltransferase genes [17]. Methylatransferases are also found in all surveyed species of a related genus, Spiroplasma [52]. These lines of evidence suggest that methyltransferases are present in the ancestors of *M. genitalium* and *M. pneumoniae*.

Second, the methyltransferase-encoding *M. pulmonis* genome is even more deficient in CpG and lower in genomic GC% than *M. genitalium* or *M. pneumoniae*, consistent with the methylation hypothesis (Fig. 26.4). It is now easy to understand that, after the loss of methyltransferase in the ancestor of *M. genitalium* and *M. pneumoniae* (Fig. 26.4), both genomes would begin to accumulate CpG dinucleotides and increase their genomic GC%. However, the evolutionary rate is much faster in *M. pneumoniae* than in *M. genitanlium* based on the comparison of a large number of protein-coding genes [85]. So *M. pneumoniae* regained CpG dinucleotide and genomic GC% much faster than *M. genitalium*. In short, the Mycoplasma data that originally seem to contradict the methylation hypothesis actually provide strong support for the methylation hypothesis when phylogeny-based genomic comparisons are made.

One might note that *Ureaplasma urealyticum* in Fig. 26.4 is not deficient in CpG because its $P_{CpG}/(P_C P_G)$ ratio is close to 1, yet its genomic GC% is the lowest. Has its low genomic GC% resulted from CpG-specific DNA methylation? If yes, then why doesn't the genome exhibit CpG deficiency? It turns out that *U. urealyticum* has C-specific, but not CpG-specific, methyltransferase, i.e., the genome of *U. urealyticum* is therefore expected to have low CG% (because of the methylation-mediated $C \rightarrow T$ mutation) but not a low $P_{CpG}/(P_C P_G)$ ratio. The methyltransferase gene from *U. urealyticum* is not homologous to that from *M. pulmonis*.

26.4 Comparative Genomics and Comparative Methods for Discrete Characters

A genome typically encodes many genes. The presence or absence of certain genes, certain phenotypic traits and environmental conditions jointly represent a major source of data for comparative genomic analysis. These binary data are best analyzed by comparative methods for discrete data.

A total of 896 bacterial genomes and 63 archaea genomes have been made available for research through Entrez as of May 21, 2009. In addition to genomic GC that can be computed as soon as the sequences are available, each sequencing project also delivers a list of genes in the sequenced genome, identified by one of two categories of methods, i.e., by checking against the 'gene dictionary' through homology search, e.g., BLAST [1, 2] or by computational gene prediction, e.g., GENSCAN [13, 14]. The availability of such annotated genoes facilitates the large-scale comparative genomics illustrated in Fig. 26.5.

The comparison in Fig. 26.5, albeit in a very small scale, can immediately lead to interesting biological questions. First, *Escherichia coli* and *Klebsiella pneumoniae* have genes coding proteins for lactose metabolism, but others do not. This leads to at least three possible evolutionary scenarios. First, lactose-metabolizing function may be absent in the ancestor A (Fig. 26.5), but (1) gained along lineage B and lost in lineage F and G or (2) gained independently along lineage E and lineage H (e.g., by lateral gene transfer or LGT). The third possible scenario is that the function is present in the ancestor A, but lost in all species except for lineages E and H.

If lactose-metabolizing genes are frequently involved in LGT, then we should expect the gene tree built from the lactose operon genes to be different from the

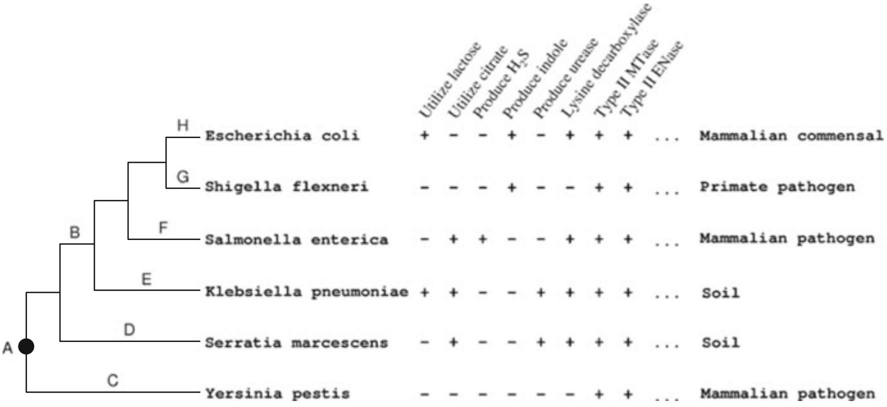

Utilize lactose
Utilize citrate
Produce H$_2$S
Produce indole
Produce urease
Lysine decarboxylase
Type II MTase
Type II ENase
H
Escherichia coli + − − + − + + + ... Mammalian commensal
G
Shigella flexneri − − − + − − + + ... Primate pathogen
F
Salmonella enterica − + + − − + + + ... Mammalian pathogen
B
E
Klebsiella pneumoniae + + − − + + + + ... Soil
D
Serratia marcescens − + − − + + + + ... Soil
A
C
Yersinia pestis − − − − − − + + ... Mammalian pathogen

Fig. 26.5 Phylogeny-based comparative bacterial genomics, with +/− indicating the presence/absence of gene-mediated functions. Modern bacterial comparative genomics typically would have thousands of columns each representing the presence/absence of one gene function as well as many environmental variables of which only a habitat variable is shown here. Modified from Ochman et al. [54]

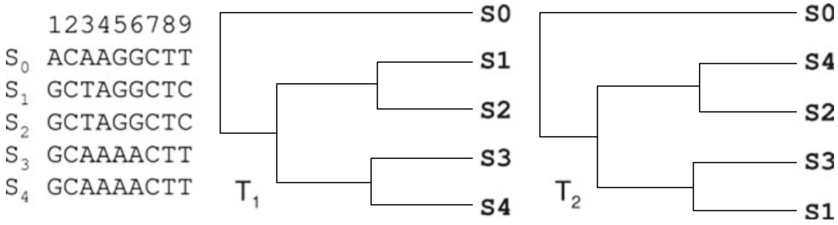


Fig. 26.6 DNA sequence data for significance tests of two alternative topologies

Table 26.4 Phylogenetic incongruence tests with maximum likelihood (ML) and maximum parsimony (MP) methods. lnL_1 and lnL_2 are site-specific log-likelihood values based on the F84 model and T_1 and T_2 (Fig. 26.6), respectively, and NC_1 and NC_2 are the minimum number of changes required for each site given T_1 and T_2, respectively

Site	ML		MP	
	lnL_1	lnL_2	NC_1	NC_2
1	−4.0975	−4.0990	1	1
2	−2.0634	−2.7767	0	0
3	−5.1147	−7.7335	1	2
4	−1.9481	−2.6238	0	0
5	−3.2142	−5.0875	1	2
6	−3.2142	−5.0875	1	2
7	−2.0634	−2.7767	0	0
8	−2.3938	−3.2626	0	0
9	−3.1090	−3.8572	1	2

species tree, which is typically approximated by a tree built from many housekeeping genes. Is the lactose operon gene tree significantly different from the species tree?

Suppose we have the sequence data (Fig. 26.6) from housekeeping genes, a species tree (T_1) and a lactose operon gene tree (T_2). We wish to test whether T_1 is significantly better than T_2 given the housekeeping gene sequences, with the null hypothesis being that T_2 is just as good as T_1. Both the maximum parsimony (MP) and the maximum likelihood (ML) methods have been used for such significance tests.

For the ML method, we compute the log-likelihood (lnL) for each of the nine sites (Fig. 26.6) given T_1 and T_2, respectively ($\mathrm{lnL_1}$ and $\mathrm{lnL_2}$ for T_1 and T_2, respectively, Table 26.4). A simple numerical illustration of computing site-specific lnL can be found in Xia [66, pp. 279–280]. A paired-sample t-test can then be applied to test whether mean $\mathrm{lnL_1}$ is significantly different from mean $\mathrm{lnL_2}$. For our data in Table 26.4, $t = 4.107$, $DF = 8$, $p = 0.0034$, two-tailed test). So we reject the null hypothesis and conclude that the lactose operon gene tree (T_2) is significantly worse than the species tree (T_1). A natural explanation for the phylogenetic incongruence is LGT.

For the MP method, we compute the minimum number of changes (NC) for each site given T_1 and T_2 (Fig. 26.6), respectively (NC_1 and NC_2 for T_1 and T_2, respectively, Table 26.4). A simple numerical illustration of computing site-specific NC can be found in Xia [66, pp. 272–275]. We can then perform a paired-sample t-test as before to test whether mean NC_1 is significantly smaller than NC_2, in one of three ways. The first is to use the entire nine pairs of data, which yields $t = -2.5298$, $DF = 8$, $p = 0.0353$, and a decision to reject the null hypothesis that T_1 and T_2 are equally good at the 0.05 significance level, i.e., T_1 is significantly better than T_2. Second, we may use only the five polymorphic sites in the paired-sample t-test, which would yield $t = -4$, $DF = 4$, and $p = 0.0161$. This leads to the same conclusion. The third is to use only the four informative sites which is however inapplicable in our case because we would have four NC_1 values all equal to 1 and four NC_2 values all equal to 2, i.e., the variation in the difference is zero.

When the phylogenetic incongruence test is applied to real lactose operon data, it was found that the lactose operon gene tree is somewhat compatible to the species tree, and the case for LGT is therefore not strong [74]. This suggests the possibility that the lactose operon was present in the ancestor, but has been lost in a number of descendent lineages. In contrast, the urease gene cluster, which is important for long-term pH homeostasis in the bacterial gastric pathogen, *Helicobacter pylori* [63, 92], generate genes trees significantly different from the species tree (unpublished result). This suggests that the urease gene cluster is involved in LGT and has implications in emerging pathogens. For example, many bacterial species pass through our digestive system daily, and it is conceivable that some of them may gain the urease gene cluster and become acid-resistant, with the consequence of one additional pathogen for our stomach.

The second type of biological questions one can derive from Fig. 26.5 is functional association between genes. We note that Type II ENase (restriction endonuclease) is always accompanied by the same type of MTase (methyltransferase) recognizing the same site (Fig. 26.5). Patterns like this allow us to quickly identify enzymes that are partners working in concert. ENase cuts the DNA at specific sites and defends the bacterial host against invading DNA phages. MTase modifies (methylates) the same site in the bacterial genome to prevent ENase from cutting the bacterial genome. Obviously, ENase activity without MTase is suicidal, so the two must both be present. This also explains why the activity of many ENases depends on S-adenosylmethionine (AdoMet) availability. AdoMet always serves as the methyl donor for MTase. Without AdoMet, the restriction sites in the host genome will not be modified even in the presence of MTase because of the lack of the methyl donor, and ENase activity will then kill the host. So it is selectively advantageous for ENase activity to depend on the availability of AdoMet. Although rare, MTase can be present without the associated ENase. For example, *E. coli* possesses two unaccompanied MTases, Dam and Dcm. Some bacteriophages carry one or more MTases to modify their own genome so as to nullify the hostile action of the host ENases.

Sometimes one may find the presence of orthologous genes in different species but the function associated with the gene is missing in some species. Such is the case of ERG genes involved in sterol metabolism. Many species, including *Drosophila melanogaster* and *Caenorhabditis elegans*, share orthologous genes involved in de novo sterol synthesis [78], but *D. melanogaster* and *C. elegans* have lost their ability to synthesize sterols de novo, although their ERG orthologs are still under strong purifying selection revealed by a much lower nonsynonymous substitution rate than the synonymous substitution rate. Further microarray studies demonstrated a strong association between the orthologs of ERG24 and ERG25 in *D. melanogaster* and genes involved in ecdysteroid synthesis and in intracellular protein trafficking and folding [78]. This suggests that the ERG genes in *D. melanogaster* have diverged and evolved new functions.

Another example in which a phylogenetic backdrop facilitates the study of evolutionary mechanisms involves the translation initiation. All molecular biology textbooks tell us that prokaryotes use the matching of the Shine-Dalgarno (SD) sequence in the mRNA and the anti-SD sequence in the small subunit rRNA to locate the translation initiation site, whereas eukaryotes use the Kozak initiation consensus to locate the translation initiation site. This would constitute a great piece of evidence for creationists to argue for independent creation. However, it is possible that the ancient organisms may have evolved these two translation initiation recognition mechanisms in parallel, and both might have contributed to the accurate localization of the translation initiation site. It is remarkable that some ancient lineages of prokaryotes living in deep sea hydrothermal vents still retain both mechanisms (unpublished results).

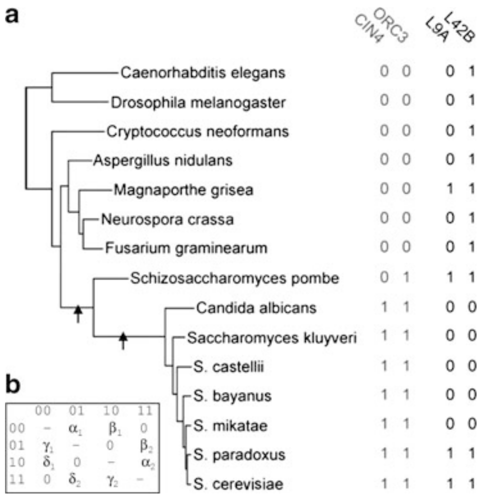
Mapping genes and gene functions to a phylogeny has revealed the loss of an essential single-copy *Maelstrom* gene in fish, and a plausible explanation is that the essential function has been fulfilled by a non-homologous gene [97]. Such findings that a specific molecular function can be performed by evolutionarily unrelated genes suggest a fundamental flaw in research effort to identify the minimal genome by identifying shared orthologous genes [47]. The rationale for such an approach is this. Suppose a minimal organism needs to perform three essential functions designated x, y, z, and three different genes, designated A, B, C, encode products that perform these three functions. If we have a genome (G1) with five genes A, B, C, D, E and another genome (G2) with four genes A, B, C, F, with genes of the same letter being orthologous, then shared orthologous genes between G1 and G2 are A, B, C which would be a good approximation of the minimal genome. In reality, it is possible that $G1 = \{A, D, E\}$ for functions x, y, z and $G2 = \{A, C, F\}$ for functions x, y, z. Both are already minimal genomes, but the intersection of G1 and G2 is only A which is a severe underestimation of a minimal genome. Creating a cell with such a 'minimal' genome is doomed to fail.

The third type of questions one can derive from Fig. 26.5 is the association between gene function and environmental variables. Note that *Klebsiella pneumoniae* and *Serratia marcescens* produce urease (Fig. 26.5). Both species can generate acids by fermentation leading to acidification of their environment. The presence of urease, which catalyzes urea to produce ammonia, can help maintain cytoplasmic pH homeostasis and allow them to tolerate environmental pH of 5 or even lower. Thus, comparative genomics can help us understand gene functions in particular environmental conditions.

Urease gene cluster serves as one of the two key acid-resistant mechanisms in the bacterial pathogen *Helicobacter pylori* in mammalian stomach, with the other mechanism being a positively charged cell membrane that alleviates the influx of protons into cytoplasm. The latter mechanism is established by comparative genomics between *H. pylori* and its close relatives as an adaptation to the acidic environment in the mammalian stomach [92].

The second and the third type of questions involve the same statistical problem, i.e., the identification of association either between two genes (e.g., between a type II ENase and a type II MTase) or between a gene and an environmental variable (e.g., between urease production and the habitat). A statistician without biological background might use a 2 × 2 contingency table (i.e., $N_{+/+}$, $N_{+/-}$, $N_{-/+}$, $N_{-/-}$) and Fisher's exact test to identify the association between two columns without taking the phylogeny into consideration. However, such an approach can lead to both false negatives and false positives. Fig. 26.7 illustrates the association study of two pairs of genes. Ignoring the phylogeny will lead to a significant association between genes *ORC3* and *CIN3*. However, the data points are not independent as the superficial association could be caused by only two consecutive gene–gain events (Fig. 26.7) and all the seven '11' could then the consequence of shared ancestral characters.

A phylogeny-based comparative analysis [7, 55] characterizes the state transition by a Markov chain, and uses a likelihood ratio test to detect the presence of


a
CIN4
ORC3
L9A
L42B
Caenorhabditis elegans 0 0 0 1
Drosophila melanogaster 0 0 0 1
Cryptococcus neoformans 0 0 0 1
Aspergillus nidulans 0 0 0 1
Magnaporthe grisea 0 0 1 1
Neurospora crassa 0 0 0 1
Fusarium graminearum 0 0 0 1
Schizosaccharomyces pombe 0 1 1 1
Candida albicans 1 1 0 0
Saccharomyces kluyveri 1 1 0 0
S. castellii 1 1 0 0
S. bayanus 1 1 0 0
S. mikatae 1 1 0 0
S. paradoxus 1 1 1 1
S. cerevisiae 1 1 1 1
b

	00	01	10	11
00	–	α_1	β_1	0
01	γ_1	–	0	β_2
10	δ_1	0	–	α_2
11	0	δ_2	γ_2	–

Fig. 26.7 Comparative methods for discrete binary characters. The presence and absence (designated by 1 and 0, respectively) of four genes are recorded for each species (**a**). The two black arrows indicate a gene–gain event. The instantaneous rate matrix (**b**), with notations following Felsenstein [22], shows the relationship among the four character designation, i.e., 00 for both genes absent, 01 for the absence of gene 1 but presence of gene 2, 10 for the presence of gene 1 but absence of gene 2, and 11 for both genes present. The diagonals are constrained by each row sum equal to 0. Modified from Barker and Pagel [7]

association between genes or between a gene function and an environmental condition. Two genes, each with two states (presence/absence), have four possible joint states and eight rate parameters ($\alpha_1, \alpha_2, \beta_1, \beta_2, \delta_1, \delta_2, \gamma_1$ and γ_2) to be estimated from the data (Fig. 26.7). When the gain or loss of one gene is independent of the other gene, then $\alpha_1 = \alpha_2, \beta_1 = \beta_2, \delta_1 = \delta_2,\ and\ \gamma_1 = \gamma_2$, with only four rate parameters to be estimated. Thus, we compute the log-likelihood for the eight-parameter and the four-parameter model given the tree and the data, designated lnL_8 and lnL_4, respectively, and perform a likelihood ratio test with test statistic being $2(lnL_8 - lnL_4)$ and four degrees of freedom.

I illustrate the computation of lnL_8 by using a simpler tree with only four operational taxonomic units or OTUs (Fig. 26.8). The joint states, represented by binary numbers 00, 01, 10 and 11, correspond to decimal numbers 0, 1, 2 and 3 which will be used to denote the four states in some equations below. The likelihood for the eight-parameter model is

$$L_8 = \sum_{z=0}^{3} \sum_{y=0}^{3} \sum_{x=0}^{3} \pi_z P_{zx}(b_6) P_{x0}(b_1) P_{x3}(b_2) P_{zy}(b_5) P_{y0}(b_3) P_{y3}(b_4) \qquad (26.8)$$

Equation 26.8 may seem to suggest that we need to sum 3^4 terms. However, the amount of computation involved is greatly reduced by the pruning algorithm [19]. To implement this algorithm, we define a vector L with elements L(0), L(1), L(2), and L(3) for every node including the leaves. L for leaf *i* is defined as

$$L_i(s) = \begin{cases} 1, & if\ s = S_i \\ 0, & otherwise \end{cases} \qquad (26.9)$$

L for an internal node with two offspring (o_1 and o_2) is recursively defined as

$$L_i(s) = \left[\sum_{k=0}^{3} P_{sk}(b_{i,o_1}) L_{o_1}(k) \right] \left[\sum_{k=0}^{3} P_{sk}(b_{i,o_2}) L_{o_2}(k) \right] \qquad (26.10)$$

where b_{i,o_1} means the branch length between internal node i and its offspring o_1, and P_{sk} is the transition probability from state s to state k computed from the rate matrix (Fig. 26.7b). For example, b_{x,S_1} (branch length between internal node x and its offspring S_1) is b_1 in Fig. 26.8. The computation involves finding the eight rate parameters that maximize L_8. As there is no analytical solution, the maximizing algorithm will simply try various rate parameter values and evaluate L_8 repeatedly until we converge on a set of parameter values that result in maximum L_8. Many such algorithms are well explained and readily available in source code [60].

While the equations might be confusing to some, the actual computation is quite simple. With only four OTUs, $S_1 = S_3 =$ '00' and $S_2 = S_4 =$ '11' (Fig. 26.8), the likelihood surface is quite flat and many different combination of the rate parameters can lead to the same maximum L_8. In fact, the only constraint on the rate parameters is high rates from states 01 and 10 to states 00 and 11 (i.e., large $\delta_1 + \gamma_1 + \alpha_2 + \beta_2$) and low rates from states 00 and 11 to states 01 and 10 (i.e., small $\delta_2 + \gamma_2 + \alpha_1 + \beta_1$). This should be obvious when we look at the four OTUs in the tree (Fig. 26.8), with only 00 and 11 being observed at the leaves. This implies that 01 and 10 should be transient states, quickly changing to 00 or 11, whereas 00 and 11 are relatively conservative stable states. One of the rate matrices that approaches the maximum L_8 is

$$Q = \left[\begin{array}{lcccc} & 00 & 01 & 10 & 11 \\ 00 & -16.47 & 13.15 & 3.32 & 0 \\ 01 & 1.10 & -135653.97 & 0 & 135652.87 \\ 10 & 1816.49 & 0 & -20308.04 & 18491.54 \\ 11 & 0 & 18.30 & 207.21 & -225.52 \end{array}\right] \tag{26.11}$$

The rate of transition from states 01 and 10 to states 00 and 11 is 644.5 times greater (The true rate should be infinitely greater) than the other way round, which implies that we will almost never observe 01 and 10 states. The transition probability matrices with branch lengths of 0.1 and 0.3, which are computed as e^{Qt}, where t is

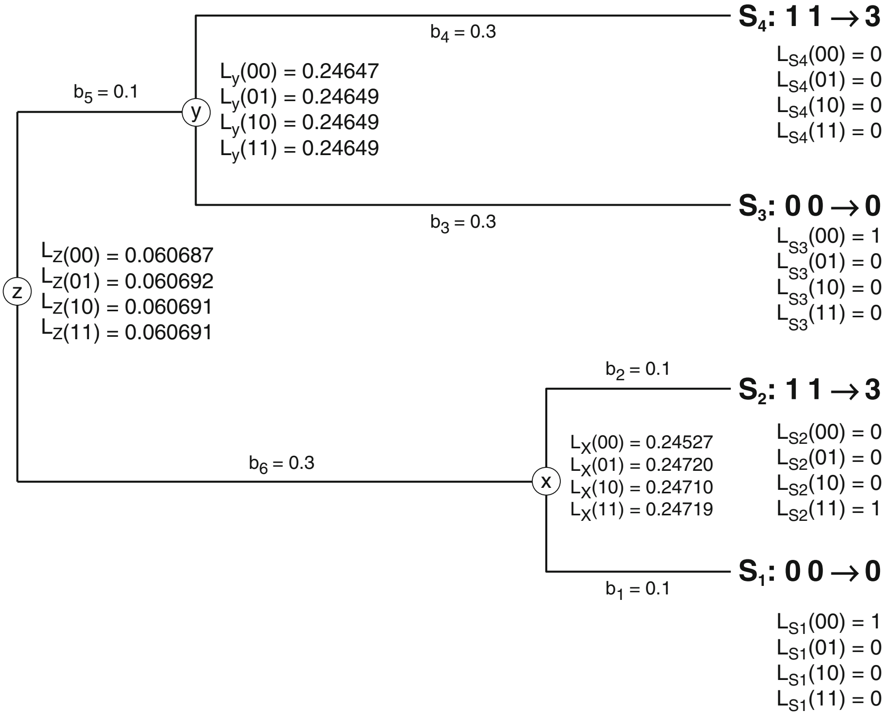


Fig. 26.8 Four-OTU tree with branch lengths (b_1–b_6) for illustrating likelihood computation. The L vectors are computed recursively according to (10)–(11)

the branch length, are, respectively,

$$P(0.1) = \left[\begin{array}{ccccc} & 00 & 01 & 10 & 11 \\ 00 & 0.54616 & 0.00011 & 0.00467 & 0.44908 \\ 01 & 0.51459 & 0.00011 & 0.00499 & 0.48038 \\ 10 & 0.51738 & 0.00011 & 0.00496 & 0.47759 \\ 11 & 0.51458 & 0.00011 & 0.00499 & 0.48034 \end{array}\right]$$
$$P(0.3) = \left[\begin{array}{ccccc} & 00 & 01 & 10 & 11 \\ 00 & 0.53145 & 0.00011 & 0.00482 & 0.46377 \\ 01 & 0.53144 & 0.00011 & 0.00482 & 0.46382 \\ 10 & 0.53144 & 0.00011 & 0.00482 & 0.46382 \\ 11 & 0.53144 & 0.00011 & 0.00482 & 0.46382 \end{array}\right] \tag{26.12}$$

We can now compute L_8 by using the pruning algorithm. First, L_{S1}–L_{S4} are straightforward from (26.9) and shown in Fig. 26.8. L_x and L_y are computed according to (26.10), e.g.,

$$\begin{aligned} L_x(00) &= P_{00,00}(0.1)P_{00,11}(0.1) = 0.54616 \times 0.44908 = 0.24527 \\ L_x(01) &= 0.51459 \times 0.48038 = 0.24720 \\ L_x(10) &= 0.51738 \times 0.47759 = 0.24710 \\ L_x(11) &= 0.51458 \times 0.48037 = 0.24719 \end{aligned} \tag{26.13}$$

Similarly, $L_y(00)$, $L_y(01)$, $L_y(10)$, and $L_y(11)$ are computed the same way and have values 0.24647, 0.24649, 0.24649, and 0.24649, respectively. Similarly, L_z is also computed by applying (26.9), e.g.,

$$\begin{aligned} L_z(00) = AB &= 0.246207 \times 0.246487 = 0.060687, \textit{ where} \\ A &= [P_{00,00}(b_6)L_x(00) + P_{00,01}(b_6)L_x(01) + P_{00,10}(b_6)L_x(10) \\ &\quad + P_{00,11}(b_6)L_x(11)] = 0.246207 \\ B &= [P_{00,00}(b_5)L_y(00) + P_{00,01}(b_5)L_y(01) + P_{00,10}(b_5)L_y(10) \\ &\quad + P_{00,11}(b_5)L_y(11)] = 0.246487 \end{aligned} \tag{26.14}$$

$L_z(01)$, $L_z(10)$, and $L_z(11)$ are 0.060692, 0.060691, and 0.060691, respectively. The final L_8 is

$$\begin{aligned} L_8 &= \sum_{k=0}^{3} \pi_k L_z(k) = 0.060687 \times 0.5 + 0.060691 \times 0.5 = 0.060689 \\ \ln(L_8) &= -2.802 \end{aligned} \tag{26.15}$$

where we used the empirical frequencies for π_k, although π_k could also be estimated as a parameter of the model. Note that states 01 and 10 are not observed, and π_{01} and π_{10} are assumed to be 0 in (26.15).

The computation of $ln(L_4)$ is simpler because only four rate parameters need to be estimated, and is equal to -5.545. If quite a large number of OTUs are involved, then twice the difference between the two log-likelihood, designated $2\Delta lnL$, follows approximately the χ^2 distribution with 4 degrees of freedom. If we could assume large-sample approximation in our case, then $2\Delta lnL = 5.486$, which leads to p $= 0.241$, i.e., the eight-parameter model is not significantly better than the four-parameter model. Such a result is not surprising given the small number of OTUs.

With this phylogeny-based likelihood approach, Barker et al. [6] found that the superficial association between genes *CIN4* and *ORC3* is not significant, although Fisher's exact test ignoring the phylogeny would produce a significant association between the two genes. Similarly, genes *L9A* and *L42B* were found to be significantly associated based on the phylogeny-based likelihood approach, although Fisher's exact test ignoring the phylogeny would suggest a lack of the association. In this particular case, *L9A* and *L42B* are known to be functionally associated and *CIN4* and *ORC3* are known not be functionally associated. Ignoring the phylogeny would have produced both a false positive and a false negative. Phylogeny-based comparative methods for continuous and discrete methods have been implemented in the freely available software DAMBE [84, 95] at http://dambe.bio.uottawa.ca.

One difficulty with the comparative methods for the continuous and discrete characters is what branch lengths to use because different trees, or even the same topology with different branch lengths, can lead to different conclusions. One may need to explore all plausible trees to check the robustness of the conclusion.

Modern comparative genomic studies may often involve the functional association of thousands of genes or more. With N genes, there are $N(N-1)/2$ possible pairwise associations and $N(N-1)/2$ tests of associations. There are $N(N-1)(N-2)/6$ possible triplet associations. So it is necessary to consider the topic of how to control for error rates in multiple comparisons.

26.5 Controlling for Error Rate in Multiple Comparisons

There are two approaches for adjusting type I error rate involving multiple comparisons, one controlling for familywise error rate (FWER), and the other controlling for the false discovery rate (FDR) [51]. While FWER methods are available in many statistical packages and covered in many books, there are few computational tutorials for the FDR in comparative genomics, an imbalance which I will try to compensate below.

The difference between the FDR and FWER is illustrated in Table 26.5, where N_{12} denotes the number of null hypotheses that are true but rejected (false positives). FWER is the probability that N_{12} is greater or equal to 1, whereas FDR is the expected proportion of $N_{12}/N_{.2}$, and defined to be 0 when $N_{.2} = 0$. Thus, FDR is a less conservative protocol for comparison, with greater power than FWER, but at a cost of increasing the likelihood of obtaining type I errors.

Table 26.5 Cross-classification of N tests of hypothesis

H_0	Reject	
	No	Yes
TRUE	N_{11}	N_{12}
FALSE	N_{21}	N_{22}
Subtotal	$N_{.1}$	$N_{.2}$

Table 26.6 Illustration of the BH [8] and BY [9] procedures in controlling for FDR, with 15 sorted p values taken from Benjamini and Hochberg [8]

i	p	$p_{critical.BH.i}$	$p_{critical.BY.i}$
1	0.0001	0.00333	0.00100
2	0.0004	0.00667	0.00201
3	0.0019	0.01000	0.00301
4	0.0095	0.01333	0.00402
5	0.0201	0.01667	0.00502
6	0.0278	0.02000	0.00603
7	0.0298	0.02333	0.00703
8	0.0344	0.02667	0.00804
9	0.0459	0.03000	0.00904
10	0.324	0.03333	0.01005
11	0.4262	0.03667	0.01105
12	0.5719	0.04000	0.01205
13	0.6528	0.04333	0.01306
14	0.759	0.04667	0.01406
15	1	0.05000	0.01507

The FDR protocol works with a set of p values. For example, with 10 genes, there are 45 pairwise tests of gene associations, yielding 45 p values. The FDR protocol is to specify a reasonable FDR (typically designated by q) and find a critical p (designated $p_{critical}$) so that a p value that is smaller than $p_{critical}$ is considered as significant, otherwise it is not. The q value is typically 0.05 or 0.01. Two general FDR procedures, Benjamini-Hochberg (BH) and Benjamini-Yekutieli (BY), are illustrated below.

Suppose we have a set of 15 sorted p values from testing 15 different hypotheses (Table 26.6). The Bonferroni method uses α /m (where m is the number of p values) as a critical p value ($p_{critical.Benferroni}$) for controlling for FWER. We have m = 15. If we take $\alpha = 0.05$, then $p_{critical.Benferroni} = 0.05/15 = 0.00333$ which would reject the first three hypotheses with the three smallest p values.

The classical FDR approach [8], now commonly referred to as the BH procedure, computes $p_{critical.BH.i}$ for the ith p value (where the subscript BH stands for the BH procedure) as

$$p_{critical.BH.i} = \frac{q \cdot i}{m} \tag{26.16}$$

where q is FDR (e.g., 0.05), and i is the rank of the p value in the sorted array of p values (Table 26.6). If k is the largest i satisfying the condition of $p_i \leq p_{critical.BH.i}$, then we reject hypotheses from H_1 to H_k. In Table 26.6, k = 4 and we reject the first four hypotheses. Note that the fourth hypothesis was not rejected by $p_{critical.Bonferroni}$ but rejected by $p_{critical.BH.4}$. Also note that $p_{critical.Bonferroni}$ is the same as $p_{critical.BH.1}$.

The FDR procedure above assumes that the test statistics are independent. A more conservative FDR procedure has been developed that relaxes the independence assumption [9]. This method, now commonly referred to as the BY procedure, computes $p_{critical.BY.i}$ for the i_{th} hypothesis as

$$p_{critical.BY.i} = \frac{q \cdot i}{m \sum_{i=1}^{m} \frac{1}{i}} = \frac{p_{critical.BH.i}}{\sum_{i=1}^{m} \frac{1}{i}} \tag{26.17}$$

With m $=$ 15 in our case, $\sum 1/i = 3.318228993$. Now k (the largest i satisfying $p_i \leq p_{critical.BY.i}$) is 3 (Table 26.6). Thus, only the first three hypotheses are rejected. The BY procedure was found to be too conservative and several alternatives have been proposed [25]. For large m, $\sum 1/i$ converges to $ln(m) + \gamma$ (Euler's constant equal approximately to 0.57721566). Thus, for m $= 10,000$, $\sum 1/i$ is close to 10. So $p_{critical.BY}$ is nearly 10 times smaller than $p_{critical.BH}$.

One may also obtain empirical distribution of p values by resampling the data. For studying association between genes or between gene and environmental factors, one may compute the frequencies of states 0 (absence) and 1 (presence) for each gene (designated f_0 and f_1, respectively) and reconstitute each column by randomly sampling from the pool of states with f_0 and f_1. For each resampling, we may carry out the likelihood ratio test shown above to obtain p values. If we have generated 10,000 p values, then the 500th smallest p value may be taken as the critical p value. Note that all the null hypotheses from resampled data are true. So FDR and FWER are equivalent. This is easy to see given that FDR is defined as the expected proportion of $N_{12}/N_{.2}$ (Table 26.5) and FWER as the probability that N_{12} (Table 26.5) is greater or equal to 1. As we cannot observe N_{ij}, we use n_{ij} to indicate their realized values. When all null hypotheses are true, $n_{22} = 0$ and $n_{12} = n_{.2}$. Now if $n_{12} > 0$, then FDR $=$ E($n_{12}/n_{.2}$) $= 1$, and FWER $=$ P($n_{12} \geq 1$) is naturally also 1. If $n_{12} = 0$, then FDR $= 0$ (Recall that FDR is defined to be 0 when $n_{.2} = 0$), and FWER $=$ P($n_{12} \geq 1$) is also 0 [8])

26.6 Comparative Viral Genomics: Detecting Viral Recombination

There are two major reasons to study recombination. The first is that it is biologically interesting. For example, different strains of viruses often recombine to form new strains of recombinants leading to host-jumping or resistance to antiviral medicine, posing direct threat to our health. The second reason is that recombination is the source of many evils in comparative genomics and molecular evolution as it can generate rate variation among sites and among lineages and distort phylogenetic relationships [43]. We may be led astray without controlling for the effect of recombination in comparative genomic analysis.

Detecting viral recombination and mapping recombination points represent important research themes in viral comparative genomics [68]. This is often done in two different situations. The first is to address whether one particular genome (typically the one causing human health concerns, designated hereafter as R) is the result of viral recombination from a set of N potential parental strains (designated hereafter as P_i, where i $= 1, 2, \ldots,$ N). Graphic visualization methods such as Simplot [45] and Bootscan [67] , as well as the phylogenetic incongruence test, are often used in this first situation.

In the second situation, one does not know which one is R and which ones are P genomes. One simply has a set of genomic sequences and wishes to know whether some are recombinants of others. This is a more difficult problem. Many methods have been developed to solve the problem, and have been reviewed lucidly [31]. I will include here only what has been left out in the review, i.e., the graphic methods (Simplot and Bootscan) for the first situation and the compatibility matrix methods for the second. The compatibility matrix methods are among the most powerful methods for detecting recombination events.

26.6.1 Is a Particular Genome a Recombinant of N Other Genomes?

Given a sequence alignment, compute genetic distances $d_{R,Pi}$ (between R and P_i) along a sliding window of typically a few hundred bases. If we have a small d_{R,P_i} and a large d_{R,P_k} for one stretch of the genome, but a large d_{R,P_i} and a small d_{R,P_k} for another stretch of the genome, then a recombination likely occurred. This method, with visualization of the d values along the sliding windows, is known as Simplot [45]. Its disadvantage is that it does not generate any measure of statistical confidence.

I will illustrate the Simplot procedure by using HIV-1M genomes in an A-J-cons-kal153.fsa file [68]. HIV-1 has three groups designated M (main), O (outgroup) and N (non-M and non-O), with the M group further divided into A-D and F-K subtypes. The A-J-cons-kal153.fsa contains consensus genomic sequences for subtypes A, B, C, D, F, G, H, and J, as well as the KAL153 strain which may be a recombinant of two of the subtypes.

The result of applying the Simplot procedure is shown in Fig. 26.9. The genetic distance used is a simultaneously estimated (SE) distance based on the F84 model [89]. Note that $d_{KAL153,A}$ is relatively small and $d_{KAL153,B}$ relatively large up to site 2,601, after which $d_{KAL153,A}$ becomes large and $d_{KAL153,B}$ small until site 8,701. After site 8,701, $d_{KAL153,A}$ again becomes small and $d_{KAL153,B}$ large (Fig. 26.9). The simplest interpretation is that KAL153 is a recombinant between an A-like strain and a B-like strain. The two sites at which KAL153 changes its phylogenetic affinity (i.e., 2,601 and 8,701) may be taken as the recombination sites.

One may ask what the interpretation would be if B is missing from the data. The interpretation unavoidably would be that KAL153 is a recombinant between

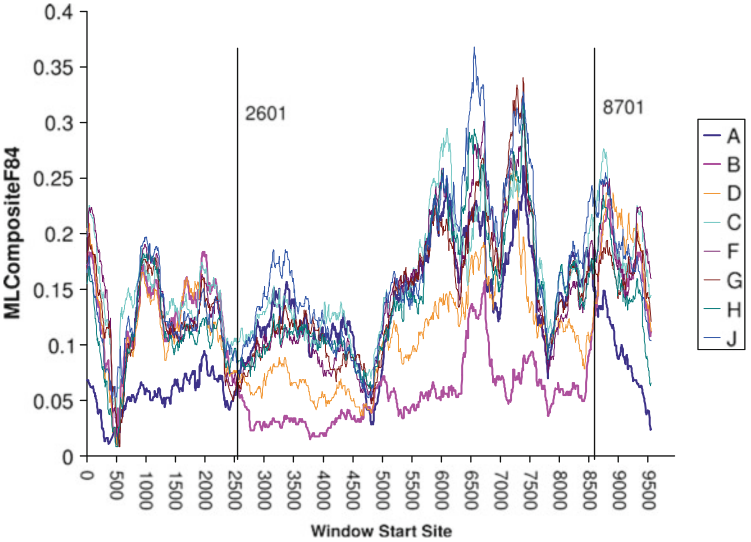


Fig. 26.9 Genetic distance between the query sequence (KAL153) and the consensus subtype sequences (A–J). MLCompositeF84 [89] is a simultaneously estimated distance based on the F84 model. KAL153 is genetically close to A before window start site at 2,601 and after window start site 8,701, but becomes close to B between window start sites 2,601 and 8,701. Output from DAMBE [84, 95]

an A-like strain and a D-like strain (Fig. 26.9). This interpretation is still reasonable because subtypes B and D are the most closely related phylogenetically. However, if A is missing from the data set, then the recombination event would become difficult to identify.

One might also note a few locations where the HIV-1 viral genomes are highly conserved across all included subtypes. Biopharmaceutical researchers typically would use such comparative genomic method to find conserved regions as drug targets or for developing vaccines against the virus.

One shortcoming of the Simplot method is that it does not produce any measure of statistical confidence. Given the stochastic nature of evolution, the distance of a sequence to other homologous sequence will often fluctuate. So the interpretation of patterns in Fig. 26.9 is associated with much uncertainty. Two approaches have been developed to overcome this shortcoming, one being the Bootscan method [67, 68], and the other is the phylogenetic incongruence test mentioned before.

The Bootscan method also takes a sliding window approach, but bootstraps the sequences to find the number of times each P_i has the smallest distance to R. The application of the bootscan method to the HIV-1M data (Fig. 26.10) shows that A is closest to KAL153 for almost all resampled data up to site 2,601, after which B becomes the closest to KAL153 until site 4,801. At this point A again becomes the closest to KAL153, albeit only briefly and with limited support. After site 5,051, B again becomes the closest to KAL153 until site 8,701 after which A again becomes

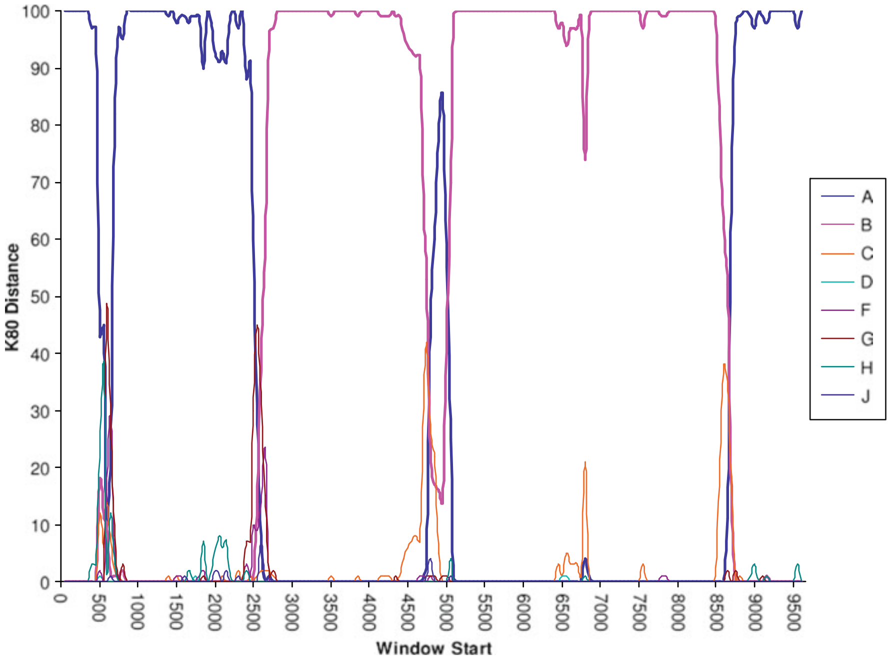


Fig. 26.10 BootScan output from scanning the HIV-1M sequences with KAL153 as the query. Output from DAMBE [84, 95], with window size being 400 nt and step size being 50 nt. DAMBE implements many other distances including the GTR distance and several simultaneously estimated distances suitable for highly diverged sequences

the closest to KAL153 (Fig. 26.10). The result suggests that there might be two recombination events.

The Simplot and the Bootscan procedures work well with highly diverged parental sequences, e.g., when the parental sequences belong to different subtypes as in our examples above. However, they are not sensitive when the parental sequences are closely related. This is true for most of the conventional methods for detecting recombination.

The second method for confirming KAL153's phylogenetic affinity reflected by changes in the genetic distance to other HIV-1M genomes (Fig. 26.9) is the phylogenetic incongruence test. The result in Fig. 26.9 allows us to partition the aligned genomic sequences into two sets, one consisting of the segment from 2,601 and 8,630 (hereafter referred to MIDDLE), and the other made of the rest of the sequences (hereafter referred to as TAILS). The phylogenetic tree for the eight subtypes of HIV-1M is shown in Fig. 26.11. A new HIV-1M genome suspected to be a recombinant, such as Kal153, may be phylogenetically grafted onto any one of the positions indicated by the numbered arrows (Fig. 26.11), creating 13 possible unrooted trees referred hereafter as T_1, T_2, ..., $T_1$3, respectively, with the subscript number corresponding to the numbers in the arrow in Fig. 26.11). From results in

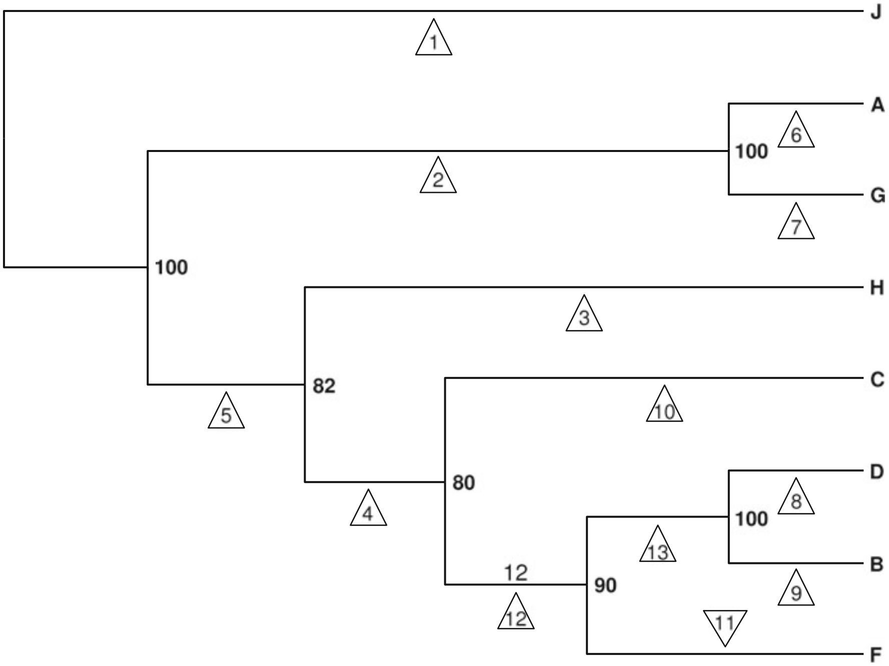


Fig. 26.11 Phylogenetic tree of the eight HIV-1M subtype genomes, with percentage bootstrap support indicated at each internal node. The numbered arrows indicate branches to which KAL153 can be granted to generate a new tree

Fig. 26.9, we can already infer that T_6 should be supported by the TAILS data set and T_9 should be supported the MIDDLE data set. However, will the support be significant against other alternative trees?

The result of phylogenetic tests (Table 26.7) shows that the TAILS data set strongly support T6 (grouping KAL153 with subtype A) but the MIDDLE data set strongly support T9 (grouping KAL153 with subtype B). This suggests that KAL153 is very highly likely to be a recombinant from subtypes A and B.

The use of the MIDDLE and TAILS for the phylogenetic incongruence test might be criticized for having fallen into a sequential testing trap [75]. A sliding-window approach together with the control for the false discover rate may be statistically more defendable.

26.6.2 General Methods Based on the Compatibility Matrix

In the set of four sequences in Fig. 26.12a, there are three possible unrooted trees labeled T_1, T_2 and T_3. Except for site 49, all sites are compatible with each other because they all support T_1. In contrast, site 49 supports T_3. In the classical population genetics with the infinite alleles model [38] where each mutation is unique and not reversible, site 49 would be considered as resulting from

Table 26.7 Statistical tests of 13 alternative trees, based on the TAILS and MIDDLE data sets

Data	Tree	*lnL*[a]	Δ*lnL*[b]	*SE*(Δ)[c]	T	*pT*[d]	*pSH*[e]	*pRELL*[f]
TAILS	6	−15046.0	0.000	0.000				1.000
	2	−15223.6	−177.587	28.579	6.214	0.000	0.000	0.000
	7	−15225.4	−179.382	28.092	6.385	0.000	0.000	0.000
	1	−15279.4	−233.325	34.684	6.727	0.000	0.000	0.000
	5	−15287.2	−241.162	34.013	7.090	0.000	0.000	0.000
	3	−15334.1	−288.028	38.281	7.524	0.000	0.000	0.000
	4	−15341.0	−294.930	38.052	7.751	0.000	0.000	0.000
	10	−15373.2	−327.121	40.059	8.166	0.000	0.000	0.000
	12	−15379.0	−332.934	39.987	8.326	0.000	0.000	0.000
	11	−15423.2	−377.209	42.205	8.938	0.000	0.000	0.000
	13	−15424.7	−378.629	41.968	9.022	0.000	0.000	0.000
	9	−15592.2	−546.125	48.274	11.313	0.000	0.000	0.000
	8	−15598.1	−552.052	47.741	11.563	0.000	0.000	0.000
MIDDLE	9	−23875.2	0.000	0.000				1.000
	13	−24086.1	−210.934	30.721	6.866	0.000	0.000	0.000
	8	−24091.5	−216.388	30.005	7.212	0.000	0.000	0.000
	12	−24398.1	−522.909	47.870	10.924	0.000	0.000	0.000
	10	−24535.3	−660.101	54.873	12.030	0.000	0.000	0.000
	4	−24553.5	−678.299	54.061	12.547	0.000	0.000	0.000
	3	−24623.9	−748.766	56.714	13.202	0.000	0.000	0.000
	5	−24627.3	−752.148	56.671	13.272	0.000	0.000	0.000
	1	−24652.2	−776.994	57.503	13.512	0.000	0.000	0.000
	2	−24653.3	−778.099	57.767	13.470	0.000	0.000	0.000
	7	−24749.9	−874.732	61.169	14.300	0.000	0.000	0.000
	6	−24753.4	−878.281	61.246	14.340	0.000	0.000	0.000

[a] log-likelihood of each tree.

[b] differences in log-likelihood between tree i and the best tree.

[c] standard error of ΔlnL.

[d] P value for paired-sample t-test (two-tailed).

[e] P value with multiple-comparison correction [72].

[f] RELL bootstrap proportions [39].

recombination because mutations, being unique and not reversible by definition with the infinite alleles model, could not produce the pattern in site 49. In other words, parallel convergent mutations in different evolutionary lineages (homoplasies) are not allowed in the infinite allele model.

The infinite alleles model is not applicable to nucleotide sequences where each site has only four possible states that can all change into each other. So we need to decide whether site 49 in Fig. 26.12a can be generated by substitutions without involving recombination. In general, sequence-based statistical methods for detecting recombination share one fundamental assumption (or flaw) that we have only two alternatives, homoplasy or recombination, to explain polymorphic site patterns in a set of aligned sequences. If we reject the homoplasy explanation, then we arrive at the conclusion of recombination which is aptly named a backdoor conclusion [29]. Such a backdoor conclusion is ultimately not as satisfying as empirical demonstrations of recombination. For example, statistical detection of

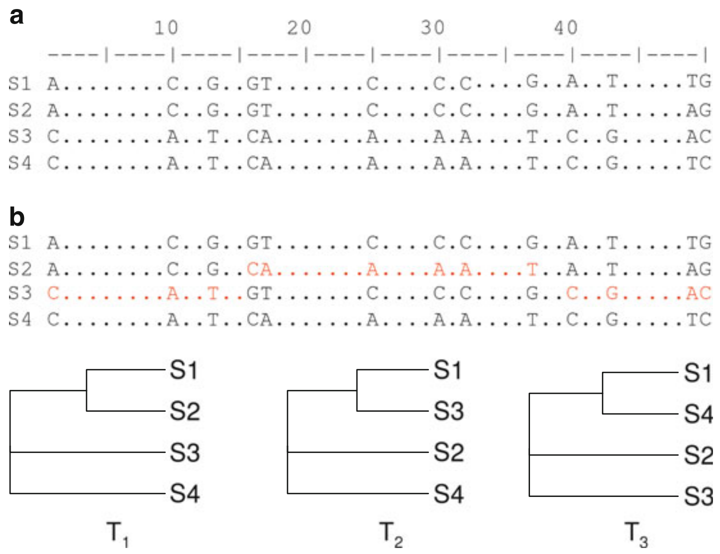

a
10 20 30 40
S1 A........C..G..GT......C....C.C....G..A..T.....TG
S2 A........C..G..GT......C....C.C....G..A..T.....AG
S3 C........A..T..CA......A....A.A....T..C..G.....AC
S4 C........A..T..CA......A....A.A....T..C..G.....TC
b
S1 A........C..G..GT......C....C.C....G..A..T.....TG
S2 A........C..G..CA......A....A.A....T..A..T.....AG
S3 C........A..T..GT......C....C.C....G..C..G.....AC
S4 C........A..T..CA......A....A.A....T..C..G.....TC
S1
S2
S3
S4
T_1
S1
S3
S2
S4
T_2
S1
S4
S2
S3
T_3

Fig. 26.12 Two sets of aligned nucleotide sequences for illustrating the compatibility-based method for detecting recombination events. (**a**) Four sequences without recombination. (**b**) Four sequences with recombination between S2 and S3, indicated by the switching of colored nucleotides. Dots indicate monomorphic sites

recombination involving mammalian mitochondrial genomes have been reported numerous times, but only an empirical demonstration [41] convinced the skeptical majority.

If we are happy with the fundamental assumption above that we have only two alternatives to discriminate between, then the method based on a compatibility matrix is both powerful and computationally fast. With a set of aligned sequences, two sites are compatible if and only if they both support the same tree topology. We only need to consider informative sites, i.e., sites featuring at least two states each of which is represented by at least two sequences. Non-informative sites are always compatible with other sites and need not be considered.

A pairwise compatibility matrix, or just compatibility matrix for short, lists whether sites i and j are compatible. The compatibility matrices for the two set of sequences in Fig. 26.12, one experiencing no recombination (Fig. 26.12a) and the other experiencing recombination involving the segment between informative sites 16–39 (Fig. 26.12b) are shown in Table 26.8. Two points are worth highlighting. First, sites that share the same evolutionary history are expected to be more compatible than those that do not (e.g., when the shared ancestry is disrupted by recombination). Note more 0's (compatible sites) in the upper triangle for sequences without recombination than in the lower triangle for sequences with recombination involving informative sites 16–39 (Table 26.8). Second, recombination tends to create similar neighbors in the compatibility matrix. Note the blocks of 1's and 0's in the lower triangle in Table 26.8. This similarity among neighbors has been

Table 26.8 Pairwise compatibility matrices, with 0 for compatible sites and 1 for incompatible sites, for aligned sequences in Fig. 26.12a (*upper triangle*) without recombination and those in Fig. 26.12b (*lower triangle*) with recombination between informative sites 16–39

Site	1	10	13	16	17	25	30	32	37	40	43	49	50
1		0	0	0	0	0	0	0	0	0	0	1	0
10	0		0	0	0	0	0	0	0	0	0	1	0
13	0	0		0	0	0	0	0	0	0	0	1	0
16	1	1	1		0	0	0	0	0	0	0	1	0
17	1	1	1	0		0	0	0	0	0	0	1	0
25	1	1	1	0	0		0	0	0	0	0	1	0
30	1	1	1	0	0	0		0	0	0	0	1	0
32	1	1	1	0	0	0	0		0	0	0	1	0
37	1	1	1	0	0	0	0	0		0	0	1	0
40	0	0	0	1	1	1	1	1	1		0	1	0
43	0	0	0	1	1	1	1	1	1	0		1	0
49	1	1	1	1	1	1	1	1	1	1	1		1
50	0	0	0	1	1	1	1	1	1	0	0	1	

characterized by the neighbor similarity score (NSS) which is the fraction of neighbors sharing either 0 (compatible) or 1 (incompatible). NSS is the basis of a number of methods for detecting recombination events [11,34,58,59,81] because its significance can be easily assessed by reshuffling the sites and recomputing NSS many times. The clumping of the compatible and incompatible sites in the compatibility matrix also suggests the possibility of mapping the recombination points. For example, one may infer from the compatibility matrix for the four sequences in Fig. 26.12b (lower triangle in Table 26.8) that the 5'-end recombination point is between informative sites 13 and 16, and that the 3'-end recombination point is between informative sites 37 and 40.

The compatibility matrix approach can be refined in two ways. First, when sequences are many, one will have some sites that are highly incompatible with each other as well as some sites that are only slightly incompatible with each other. The compatibility matrix approach lumps all these sites as incompatible sites, resulting in loss of information. Second, neighboring sites in a set of aligned sequences are expected to be more compatible with each other than with sites that are far apart. These two refinements were included in a recent study [12] that uses a refined incompatibility score (RIS) and the PHI statistic based on RIS. This new method appears much more sensitive than previous ones based on empirical applications [12,65].

26.7 Summary

With the increasing availability of genomic sequences, comparative genomics has expanded rapidly and contributed significantly to our understanding of how mutation, recombination and natural selection have jointly governed the evolutionary process. Comparative genomic analysis, aided by the phylogeny-based comparative methods, has resulted in improved detection of (1) functional association between genes and between genes and environment which is essential for understanding the origin and maintenance of the genetic components of biodiversity, (2) lateral gene transfer in prokaryotes and (3) recombination events and recombination sites. Development of comparative genomics has also motivated the research in statistics such as those controlling for the false discovery rates. Comparative genomics has dramatically changed the way of how regulatory sequence motifs are discovered, leading to the active development of phylogenetic footprinting which will be covered in the next chapter. What is particularly worth pointing out is that powerful and sophisticated software packages have been developed to facilitate research in comparative genomics.

Acknowledgements I thank J. Felsenstein and M. Pagel for identifying ambiguities and errors in the manuscript and for their many suggestions to improve the manuscript. S. Aris-Brosou, Y. B. Fu and G. Palidwor, as well as two anonymous reviewers, provided comments and references. I am supported by the Strategic Research, Discovery and Research Tools and Instrument Grants of Natural Science and Engineering Research Council of Canada.

References

1. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., & Lipman, D.J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, *215*, 403–410.
2. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang Z., M., & Lipman, D.J. (1997). Gapped blast and psi-blast: A new generation of protein database search programs. *Nucleic Acids Research*, *25*, 3389–3402.
3. Argos, P., Rossmann, M.G., Grau, U.M., Zuber, A., Franck, G., & Tratschin, J.D. (1979). Thermal stability and protein structure. *Biochemistry (Moscow)*, *18*, 5698–5703.
4. Aris-Brosou, S., & Xia, X. (2008). Phylogenetic analyses: A toolbox expanding towards Bayesian methods. *International Journal of Plant Genomics*, *2008*, DOI 10.1155/2008/683509
5. Ballester, R., Marchuk, D., Boguski, M., Saulino, A., Letcher, R., & Wigler, M. (1990). The nf1 locus encodes a protein functionally related to mammalian gap and yeast ira proteins. *Cell*, *63*, 851–859.
6. Barker, D., Meade, A., & Pagel, M. (2007). Constrained models of evolution lead to improved prediction of functional linkage from correlated gain and loss of genes. *Bioinformatics*, *23*, 14–20.
7. Barker, D., & Pagel, M. (2005). Predicting functional gene links from phylogenetic-statistical analyses of whole genomes. *PLoS Computational Biology*, *1*, e3.
8. Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, *57*, 289–300.
9. Benjamini, Y., & Yekutieli, D. (2001). The control of the false discovery rate in multiple hypothesis testing under dependency. *The Annals of Statistics*, *29*, 1165–1188.
10. Bestor, T.H., & Coxon, A. (1993). The pros and cons of dna methylation. *Current Biology*, *6*, 384–386.

11. Brown C.J., Garner, E.C., Dunker, A.K, & Joyce, P. (2001). The power to detect recombination using the coalescent. *Molecular Biology and Evolution*, *18*, 1421–1424.
12. Bruen, T.C., Philippe, H., & Bryant, D. (2006). A simple and robust statistical test for detecting the presence of recombination. *Genetics*, *172*, 2665–2681.
13. Burge, C., & Karlin, S. (1997). Prediction of complete gene structures in human genomic dna. *Journal of Molecular Biology*, *268*, 78–94.
14. Burge, C.B., & Karlin, S. (1998). Finding the genes in genomic dna. *Current Opinion in Structural Biology*, *8*, 346–354.
15. Cardon, L.R., Burge, C., Clayton, D.A., Karlin, S. (1994). Pervasive CpG suppression in animal mitochondrial genomes. *Proceedings of the National Academy of Sciences*, *91*, 3799–3803.
16. Carullo, M., & Xia, X. (2008). An extensive study of mutation and selection on the wobble nucleotide in trna anticodons in fungal mitochondrial genomes. *Journal of Molecular Evolution*, *66*, 484–493.
17. Chambaud, I., Heilig, R., Ferris, S., Barbe, V., Samson, D., Galisson, F., et al. (2001). The complete genome sequence of the murine respiratory pathogen mycoplasma pulmonis. *Nucleic Acids Research*, *29*, 2145–2153.
18. Dalgaard, J.Z., & Garrett, R.A., (1993). Archaeal hyperthermophile genes. In M. Kates, D. J. Kushner, & A. T. Matheson (Eds.), *The biochemistry of Archaea (Archaebacteria).* Amsterdam: Elsevier.
19. Felsenstein, J. (1981). Evolutionary trees from dna sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, *17*, 368–376.
20. Felsenstein, J. (1985). Phylogenies and the comparative method. *American Natural*, *125*, 1–15.
21. Felsenstein, J. (2002). *PHYLIP 3.6 (phylogeny inference package).* Seattle: Department of Genetics, University of Washington.
22. Felsenstein, J. (2004). *Inferring phylogenies*. Sunderland, Massachusetts: Sinauer.
23. Frederico, L.A., Kunkel, T.A., & Shaw, B.R. (1990). A sensitive genetic assay for the detection of cytosine deamination determination of rate constants and the activation energy. *Biochemistry (Moscow)*, *29*, 2532–2537.
24. Galtier, N., & Lobry, J.R. (1997). Relationships between genomic g+c content, rna secondary structures, and optimal growth temperature in prokaryotes. *Journal of Molecular Evolution*, *44*, 632–636.
25. Ge, Y., Sealfon, S.C., & Speed, T.P. (2008). Some step-down procedures controlling the false discovery rate under dependence. *Statistica Sinica*, *18*, 881–904.
26. Gordon, J.L., Byrne, K.P., & Wolfe, K.H. Additions, losses, and rearrangements on the evolutionary route from a reconstructed ancestor to the modern saccharomyces cerevisiae genome. *PLoS Genetics*, *5*(5), e1000,485. DOI 10.1371/journal.pgen.1000485
27. Goto M., Washio T., Tomita M. (2000). Causal analysis of CpG suppression in the Mycoplasma genome. Microbial and Comparative Genomics, 5, 51–58.
28. Harvey, P.H., & Pagel, M.D. (1991). *The comparative method in evolutionary biology*. Oxford: Oxford University Press.
29. Hey, J. (2000). Human mitochondrial dna recombination: can it be true? *Trends in Ecology and Evolution*, *15*, 181–182.
30. Hurst, L.D., & Merchant, A.R. (2001). High guanine-cytosine content is not an adaptation to high temperature: A comparative analysis amongst prokaryotes. *Proceedings of the Royal Society B*, *268*, 493–497.
31. Husmeier, D., & Wright, F. (2005). Detectign recombination in DNA sequence alignments. In D. Husmeier, R. Dybowski, & S. Roberts (Eds.), *Probabilistic modeling in bioinformatics and medical informatics* (p. 504). London: Springer.
32. Irimia, M., Penny, D., & Roy, S.W. (2007). Coevolution of genomic intron number and splice sites. *Trends Genetics*, *23*, 321.
33. Jacob, F. (1988). *The statue within: an autobiography*. New York: Basic Books, Inc.
34. Jakobsen, I.B., & Easteal, S. (1996). A program for calculating and displaying compatibility matrices as an aid in determining reticulate evolution in molecular sequences. *Computer Applications in the Biosciences*, *12*, 291–295.

35. Josse, J., Kaiser, A.D., & Kornberg, A. (1961). Enzymatic synthesis of deoxyribonucleic acid vii. frequencies of nearest neighbor base-sequences in deoxyribonucleic acid. *The Journal of Biological Chemistry*, *236*, 864–875.
36. Karlin, S., & Burge, C. (1995). Dinucleotide relative abundance extremes: A genomic signature. *Trends in Genetics*, *11*, 283–290.
37. Karlin, S., & Mrazek, J. (1996). What drives codon choices in human genes. *The Journal of Biological Chemistry*, *262*, 459–472.
38. Kimura, M., & Crow, A.J.F (1964). The number of alleles that can be maintained in a finite population. *Genetics*, *49*, 725–738.
39. Kishino, H., & Hasegawa, M. (1989). Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from dna sequence data, and the branching order in hominoidea. *Journal of Molecular Evolution*, *29*, 170–179.
40. Kliman, R.M., & Bernal, C.A. (2005). Unusual usage of agg and ttg codons in humans and their viruses. *Gene*, *352*, 92.
41. Kraytsberg, Y., Schwartz, M., Brown, T.A., Ebralidse, K., Kunz, W.S., Clayton, D.A., et al. (2004). Recombination of human mitochondrial dna. *Science*, *304*, 981.
42. Kushiro, A., Shimizu, M., & Tomita, K. I. (1987). Molecular cloning and sequence determination of the tuf gene coding for the elongation factor tu of thermus thermophilus hb8. *European Journal of Biochemistry*, *170*, 93–98.
43. Lemey, P., & Posada, D. (2009). Introduction to recombination detection. In P. Lemey, M. Salemi, & A. M. Vandamme AM, *The phylogenetic handbook* (2nd ed.). Cambridge: Cambridge University Press.
44. Lindahl, T. (1993). Instability and decay of the primary structure of dna. *Nature*, *362*, 709–715.
45. Lole, K.S., Bollinger, R.C., Paranjape, R.S., Gadkari, D., Kulkarni, S.S., Novak, N.G., et al. (1999). Full-length human immunodeficiency virus type 1 genomes from subtype c-infected seroconverters in india, with evidence of intersubtype recombination. *The Journal of Virology*, *73*, 152–160.
46. Martins, E.P., & Hansen, T.F. (1997). Phylogenies and the comparative method: A general approach to incorporating phylogenetic information into the analysis of interspecific data. *The American Naturalist*, *149*(4), 646–667.
47. Mushegian, A.R., & Koonin, E.A. (1996). Minimal gene set for cellular life derived by comparison of complete bacterial genomes. *Proceedings of the National Academy of Sciences of the United States of America*, *93*, 10268–10273.
48. Muto, A., & Osawa, S. (1987). The guanine and cytocine content of genomic dna and bacterial evolution. *Proceedings of the National Academy of Sciences*, *84*, 166–169.
49. Nakashima, H., Fukuchi, S., & Nishikawa, K. (2003). Compositional changes in rna, dna and proteins for bacterial adaptation to higher and lower temperatures. *The Journal of Biochemistry (Tokyo)*, *133*, 507–513.
50. Nei, M., & Kumar, S. (2000). *Molecular evolution and phylogenetics*. New York: Oxford University Press.
51. Nichols, T., & Hayasaka, S. (2003). Controlling the familywise error rate in functional neuroimaging: A comparative review. *Statistical Methods in Medical Research*, *12*, 419–446.
52. Nur, I., Szyf, M., Razin, A., Glaser, G., Rottem, S., & Razin, S. (1985). Procaryotic and eucaryotic traits of dna methylation in spiroplasmas (mycoplasmas). *The Journal of Bacteriology*, *164*, 19–24.
53. Nussinov, R. (1984). Doublet frequencies in evolutionary distinct groups. *Nucleic Acids Research*, *12*, 1749–1463.
54. Ochman, H., Lawrence, J.G., & Groisman, E.A. (2000). Lateral gene transfer and the nature of bacterial innovation. *Nature*, *405*, 299–304.
55. Pagel, M. (1994). Detecting correlated evolution on phylogenies: A general method for the comparative analysis of discrete characters. *Proceedings of the Royal Society London B: Biological Sciences*, *255*, 37–45.
56. Pagel, M. (1997). Inferring evolutionary processes from phylogenies. *Zoologica Scripta*, *26*, 331–348.

57. Pagel, M. (1999). Inferring the historical patterns of biological evolution. *Nature*, *401*, 877–884.
58. Posada, D. (2002). Evaluation of methods for detecting recombination from dna sequences: Empirical data. *Molecular Biology and Evolution*, *19*, 708–717.
59. Posada, D., & Crandall, K.A. (2001). Evaluation of methods for detecting recombination from dna sequences: Computer simulations. *Proceedings of the National Academy of Sciences of the United States of America*, *98*, 13757–13762.
60. Press, W.H., Teukolsky, S.A., Tetterling, W.T., & Flannery, B.P. (1992). *Numerical recipes in C the art of scientifi computing* (2nd edn.). Cambridge: Cambridge University Press.
61. Razin, A., & Razin, S. (1980). Methylated bases in mycoplasmal dna. *Nucleic Acids Research*, *8*, 1383–1390.
62. Rideout, W.M.I., Coetzee, G.A., Olumi, A.F., & Jones, P.A. (1990). 5-methylcytosine as an endogenous mutagen in the human ldl receptor and p53 genes. *Science*, *249*, 1288–1290.
63. Sachs, G., Weeks, D.L., Melchers, K., & Scott, D.R. (2003). The gastric biology of helicobacter pylori. *Annual Review of Physiology*, *65*, 349–369.
64. Saenger, W. (1984). *Principles of nucleic acid structure*. New York: Springer.
65. Salemi, M., Gray, R.R., & Goodenow, M.M. (2008). An exploratory algorithm to identify intrahost recombinant viral sequences. *Molecular Phylogenetics and Evolution*, *49*, 618.
66. Salemi, M., & Vandamme, A.-M. (eds.) (2003). *The Phylogenetic Handbook: A Practical Approach to DNA and Protein Phylogeny*. Cambridge University Press.
67. Salminen, M.O., Carr, J.K., Burke, D.S., & McCutchan, F.E. (1995). Identification of breakpoints in intergenotypic recombinants of hiv type 1 by bootscanning. *AIDS Research and Human Retroviruses*, *11*, 1423–1425.
68. Salminen, M., & Martin, D. (2009). Detecting and characterizing individual recombination events. In P. Lemey, M. Salemi, A. M. Vandamme (Eds.), *The phylogenetic handbook* (2nd ed.). Cambridge: Cambridge University Press.
69. Sankoff, D. (2009). Reconstructing the history of yeast genomes. *PLoS Genetics*, *5*, e1000,483.
70. Sankoff, D., & El-Mabrouk, N. (2002). Genome rearrangement. In T. Jiang, Y. Xu, & M. Q. Zhang (Eds.), *Current topics in computational molecular biology*. Cambridge: MIT.
71. Schluter, D., Price, T.D., Mooers, A.Ø., & Ludwig, D. (1997). Likelihood of ancestor states in adaptive radiation. *Evolution*, *51*, 1699–1711.
72. Shimodaira, H., & Hasegawa, M. (1999). Multiple comparisons of log-likelihoods with applications to phylogenetic inference. *Molecular Biology and Evolution*, *16*, 1114–1116.
73. Singer, C.E., & Ames, B.N. (1970). Sunlight ultraviolet and bacterial dna base ratios. Science, *170*, 822–826.
74. Stoebel, D.M. (2005). Lack of evidence for horizontal transfer of the lac operon into escherichia coli. *Molecular Biology and Evolution*, *22*, 683–690.
75. Suchard, M.A., Weiss, R.E., Dorman, K.S., & Sinsheimer, J.S. (2002). Oh brother, where art thou? a bayes factor test for recombination with uncertain heritage. *The Systems Biology*, *51*, 715–728.
76. Sueoka, N. (1964). *On the evolution of informational macromolecules*. New York: Academic.
77. Sved, J., & Bird, A. (1990). The expected equilibrium of the cpg dinucleotide in vertebrate genomes under a mutation model. *Proceedings of the National Academy of Sciences of the United States of America*, *87*, 4692–4696.
78. Vinci, G., Xia, X., & Veitia, R.A. (2008). Preservation of genes involved in sterol metabolism in cholesterol auxotrophs: Facts and hypotheses. *PLoS ONE*, *3*, e2883.
79. Wang, H.C., & Hickey, D.A. (2002). Evidence for strong selective constraint acting on the nucleotide composition of 16s ribosomal rna genes. *Nucleic Acids Research*, *30*, 2501–2507.
80. Wang, H.C., Xia, X., & Hickey, D.A. (2006). Thermal adaptation of ribosomal rna genes: A comparative study. *Journal of Molecular Evolution*, *63*, 120–126.
81. Wiuf, C., Christensen, T., & Hein, J. (2001). A simulation study of the reliability of recombination detection methods. *Journal of Molecular Evolution*, *18*, 1929–1939.
82. Xia, X. (1998). How optimized is the translational machinery in escherichia coli, salmonella typhimurium and saccharomyces cerevisiae? *Genetics*, *149*, 37–44.

83. Xia, X. (1998). The rate heterogeneity of nonsynonymous substitutions in mammalian mitochondrial genes. *Journal of Molecular Evolution*, *15*, 336–344.
84. Xia, X. (2001). *Data analysis in molecular biology and evolution*. Boston: Kluwer Academic Publishers.
85. Xia, X. (2003). Dna methylation and mycoplasma genomes. *Journal of Molecular Evolution*, *57*, S21–S28.
86. Xia, X. (2005). Mutation and selection on the anticodon of trna genes in vertebrate mitochondrial genomes. *Gene*, *345*, 13–20.
87. Xia, X. (2007). Molecular phylogenetics: Mathematical framework and unsolved problems. In U. Bastolla, M. Porto, H. E. Roman, & M. Vendruscolo (Eds.), *Structural approaches to sequence evolution* (pp. 171–191).
88. Xia, X. (2008). The cost of wobble translation in fungal mitochondrial genomes: Integration of two traditional hypotheses. *BMC Evolutionary Biology*, *8*, 211.
89. Xia, X. (2009). Information-theoretic indices and an approximate significance test for testing the molecular clock hypothesis with genetic distances. *Molecular Phylogenetics and Evolution*, *52*, 665–676.
90. Xia, X., Huang, H., Carullo, M., Betran, E., & Moriyama, E.N. (2007). Conflict between translation initiation and elongation in vertebrate mitochondrial genomes. *PLoS ONE*, *2*, e227.
91. Xia, X., & Li, W.H. (1998). What amino acid properties affect protein evolution? *Journal of Molecular Evolution*, *47*, 557–564.
92. Xia, X., & Palidwor, G. (2005). Genomic adaptation to acidic environment: Evidence from helicobacter pylori. *The American Naturalist*, *166*, 776–784.
93. Xia, X., Wang, H.C., Xie, Z., Carullo, M., Huang, H., & Hickey, D.A. (2006). Cytosine usage modulates the correlation between cds length and cg content in prokaryotic genomes. *Molecular Biology and Evolution*, *23*, 1450–1454.
94. Xia, X.H, Wei, T., Xie, Z., & Antoine, D. (2002). Genomic changes in nucleotide and dinucleotide frequencies in pasteurella multocida cultured under high temperature. *Genetics*, *161*, 1385–1394.
95. Xia, X., & Xie, Z. (2001). Dambe: Software package for data analysis in molecular biology and evolution. *Journal of Heredity*, *92*, 371–373.
96. Xia, X., & Yuen, K.Y. (2005). Differential selection and mutation between dsdna and ssdna phages shape the evolution of their genomic at percentage. *BMC Genetics*, *6*, 20.
97. Zhang, D., Xiong, H., Shan, J., Xia, X., & Trudeau, V. (2008). Functional insight into maelstrom in the germline pirna pathway: A unique domain homologous to the dnaq-h 3-5 exonuclease, its lineage-specific expansion/loss and evolutionarily active site switch. *Biology Directorate*, *3*, 48.

Chapter 27
Robust Control of Immune Systems Under Noises: Stochastic Game Approach

Bor-Sen Chen, Chia-Hung Chang, and Yung-Jen Chuang

Abstract A robust control of immune response is proposed for therapeutic enhancement to match a prescribed immune response under uncertain initial states and environmental noises, including continuous intrusion of exogenous pathogens. The worst-case effect of all possible noises and uncertain initial states on the matching for a desired immune response is minimized for the enhanced immune system, i.e., a robust control is designed to track a prescribed immune model response from the stochastic minimax matching perspective. This minimax matching problem could herein be transformed to an equivalent stochastic game problem. The exogenous pathogens and environmental noises (external noises) and stochastic uncertain internal noises are considered as a player to maximize (worsen) the matching error when the therapeutic control agents are considered as another player to minimize the matching error.

Since the innate immune system is highly nonlinear, it is not easy to solve the robust control problem by the nonlinear stochastic game method directly. A fuzzy model is proposed to interpolate several linearized immune systems at different operating points to approximate the innate immune system via smooth fuzzy membership functions. With the help of fuzzy approximation method, the stochastic minimax matching control problem of immune systems could be easily solved by the proposed fuzzy stochastic game method via the linear matrix inequality (LMI)

B.-S. Chenc (✉)
Department of Electrical Engineering,National Tsing Hua University, 101, Section 2,
Kuang-Fu Road, Hsinchu, Taiwan
e-mail: bschen@ee.nthu.edu.tw

C.-H. Chang
Department of Electrical Engineering, National Tsing Hua University, 101, Section 2,
Kuang-Fu Road, Hsinchu, Taiwan
e-mail: chchang70756@gmail.com

Y.-J. Chuang
Institute of Bioinformatics and Structural Biology, National Tsing Hua University, 101,
Section 2, Kuang-Fu Road, Hsinchu, Taiwan
e-mail: yjchuang@life.nthu.edu.tw

H. Horng-Shing Lu et al. (eds.), *Handbook of Statistical Bioinformatics*,
Springer Handbooks of Computational Statistics, DOI 10.1007/978-3-642-16345-6_27,

© Springer-Verlag Berlin Heidelberg 2011

technique with the help of Robust Control Toolbox in Matlab. Finally, *in silico* examples are given to illustrate the design procedure and to confirm the efficiency and efficacy of the proposed method.

27.1 Introduction

A dynamic response of the immune system, which includes innate immune system and adaptive immune system, is induced by infectious microbes or noises. The innate immune system provides a tactical response, signaling the presence of 'non-self' organisms and activating B cells to produce antibodies to bind to the intruders' epitopic sites. The antibodies identify targets for scavenging cells that engulf and consume the microbes, reducing them to non-functioning units [42]. The antibodies can also stimulate the production of cytokines, complement factors and acute-phase response proteins that either damage an intruder's plasma membrane directly or trigger the second phase of immune response. The innate immune system protects against many extracellular bacteria or free viruses found in blood plasma, lymph, tissue fluid, or interstitial space between cells, but it cannot clean out microbes that burrow into cells, such as viruses, intracellular bacteria, and protozoa [17, 23, 42]. The innate immune system is a complex system and the obscure relationships between the immune system and the environment in which several modulatory stimuli are embedded (e.g., antigens, molecules of various origin, physical stimuli, stress stimuli).This environment is noisy because of the great amount of such signals. The immune noise has therefore at least two components: (a) the internal noise, due to the exchange of a network of molecular and cellular signals belonging to the immune system during an immune response or in the homeostasis of the immune system. The concept of the internal noise might be viewed in biological terms as a status of sub-inflammation required by the immune response to occur; (b) the external noise, the set of external signals that target the immune system (and hence that add noise to the internal one) during the whole life of an organism.

Activated by the innate immune response, the adaptive immune system could provide strategic response to invading microbe and yield protective cells. These protective cells could remember specific antigens and produce antibodies to counter the antigens, and seek for epitopes of antigens on the surfaces of infected cells. It is found that adaptive immune mechanisms depend on the actions of B- and T-lymphocytes that become dedicated to a single antibody type through clonal selection. Meanwhile, killer T-cells (or cytotoxic T-lymphocytes) bind to infected cells and kill them by initiating programmed cell death (apoptosis). In addition, helper T-cells assist naive B-cells in maturing into plasma cells that produce the needed antibody type. Finally, immune cells with narrowly focused memory are generated, ready to respond rapidly if invading microbes with the same antigen epitopes are encountered again. Elements of the innate and adaptive immune systems are shared, and response mechanisms are coupled, even though distinctive modes of operation can be identified [17, 23, 42].

Recently, there are many models of immune response to infection [3, 25, 30, 33] with special emphasis on the human-immunodeficiency virus [26, 28, 29, 39]. Some papers have discussed immune defense models with moving target strategy [1]. Norbert Wiener [48] and Richard Bellman [7] appreciated and anticipated the application of mathematical analysis to treatment in a broad sense, and Swan surveys early optimal control applications to biomedical problems [43]. Notably, Kirschner [19] offers an optimal control approach to HIV treatment, and intuitive control approaches are presented [8, 15, 47, 49, 50].

The dynamics of drug response (pharmacokinetics) have been modeled in several works [32, 45] and control theory is applied to drug delivery in other studies [6, 10, 14, 16, 18, 20, 27, 31, 34]. Recently, Stengel [42] presented a simple model for the response of the innate immune system to infection and therapeutic intervention by applying the quadratic optimal control design which finds a control agent such that the immune response is stable and the quadratic performance index (cost function) is minimized. Their results show not only the progression from an initially life-threatening state to a controlled or cured condition but also the optimal history of therapeutic agents that produces that condition. In their study, the performance index (cost function) of quadratic optimal control for immune systems may be decayed by the continuous exogenous pathogens input, which is considered as noises of the immune system. Furthermore, some overshoots may occur in the optimal control process and may lead to organ failure because the quadratic optimal control only minimizes a quadratic performance index (cost function) that is only the integration of squares of states and allows the existence of overshoot [51]. A series researches about dynamic optimization method which find a control law for immune system such that a certain optimality criterion is achieved is proposed to design the optimal schedule for host defense, immune memory and post-infection pathogen levels in mammals [35–38]. Recently, a minimax robust tracking control for immune to match the desired immune response systems under environmental disturbances has been studied [11].

In this study, a robust control of immune response is proposed for therapeutic enhancement to match a desired immune response under uncertain exogenous pathogens input, noises and uncertain initial states. Because of the uncertainties of these factors mentioned above, in order to attenuate their detrimental effects, their worst-case effects should be considered in the matching control procedure from the robust design perspective. The worst-case effect of all possible uncertain factors on the matching error to a desired immune response is minimized for the enhanced immune systems, i.e., the proposed robust control is designed from the stochastic minimax matching perspective. This minimax matching could be transformed to an equivalent dynamic game problem [5]. The exogenous pathogen input is considered as a player to maximize (worsen) the matching error, while the therapeutic control agent is considered as another player to minimize the matching error. Since the innate immune system is highly nonlinear, it is not easy to solve the robust control problem by the nonlinear stochastic game method directly. Recently, fuzzy systems have been employed to efficiently approximate nonlinear dynamic systems to solve the nonlinear control problem [12, 13, 21, 22]. A fuzzy model is proposed

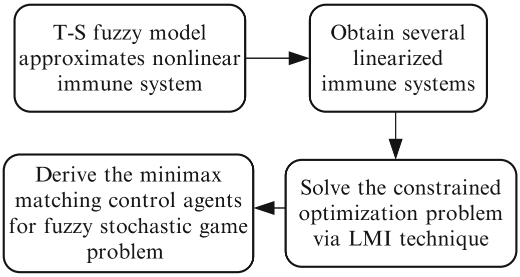

T-S fuzzy model approximates nonlinear immune system
Obtain several linearized immune systems
Solve the constrained optimization problem via LMI technique
Derive the minimax matching control agents for fuzzy stochastic game problem

Fig. 27.1 Scheme of the robust control design for innate immune systems

to interpolate several linearized immune systems at different operating points to approximate the innate immune system via smooth fuzzy membership functions. Then, with the help of fuzzy approximation method, a fuzzy dynamic game scheme is developed so that the stochastic minimax matching control of immune systems could be easily solved by the linear stochastic game method, which can be subsequently solved by a constrained optimization scheme via the linear matrix inequality (LMI) technique [9] with the help of Robust Control Toolbox in Matlab. Because the fuzzy dynamic model can approximate any nonlinear dynamic system, the proposed model matching method via fuzzy game theory can be applied to the robust control design of any model of immune system that can be Takagi-Sugeno (T-S) fuzzy interpolated. Finally, the computational simulation examples are given to illustrate the design procedure and to confirm the efficiency and efficacy of the proposed minimax match control method for immune systems. The design scheme is shown in Fig. 27.1.

27.2 Model of Innate Immune System

For the principal goals to study the general course of a disease and to clarify some observational results, a simple four-nonlinear, ordinary differential equation for the dynamic model of infectious disease is introduced as the following equations to describe rates of change of pathogen, immune cell and antibody concentrations and of an indicator of organic health [3, 41]. A more general dynamic model will be given next in sequel.

$$\begin{aligned}
\dot{x}_1 &= (a_{11} - a_{12}x_3)x_1 + b_1u_1 + d_1w_1\\
\dot{x}_2 &= a_{21}(x_4)a_{22}x_1x_3 - a_{23}(x_2 - x_2^*) + b_2u_2 + d_2w_2\\
\dot{x}_3 &= a_{31}x_2 - (a_{32} + a_{33}x_1)x_3 + b_3u_3 + d_3w_3\\
\dot{x}_4 &= a_{41}x_1 - a_{42}x_4 + b_4u_4 + d_4w_4
\end{aligned} \tag{27.1}$$

$$a_{21}(x_4) = \begin{cases} \cos(\pi\ x_4), & 0 \le x_4 \le 1/2 \\ \quad 0, & 1/2 \le x_4 \end{cases}$$

where x_1 denotes the concentration of a pathogen that expresses a specific foreign antigen; x_2 denotes the concentration of immune cells that are specific to the foreign antigen; x_3 denotes the concentration of antibodies that bind to the foreign antigen; x_4 denotes the characteristic of a damaged organ [x_4=0:healthy, x_4 $\geq$1:dead]. The combined therapeutic control agents and the exogenous inputs are described as follows: u_1 denotes the pathogen killer's agent; u_2 denotes the immune cell enhancer; u_3 denotes the antibody enhancer; u_4 denotes the organ healing factor (or health enhancer); and w_1 denotes the rate of continuing introduction of exogenous pathogens (external noises). $w_2 \sim w_4$ denote the stochastic noises or unmodeled errors and residues (internal noises). $a_{21}(x_4)$ is a nonlinear function that describes the mediation of immune cell generation by the damaged cell organ. And if there is no antigen, then the immune cell maintains the steady equilibrium value of x_2^*. The parameters have been chosen to produce a system that recovers naturally from the pathogen infections (without treatment) as a function of initial conditions during a period of times. For the benchmark example in (27.1), both parameters and time units are abstractions, as no specific disease is addressed. The state and control are always positive because concentrations cannot go below zero, and organ death is indicated when $x_4 \geq 1$. The structural relationship of system variables in (27.1) is illustrated in Fig. 27.2. Organ health mediates immune cell production, inferring a relationship between immune response and fitness of the individual. Antibodies bind to the attacking antigens, thereby killing pathogenic microbes directly, activating complement proteins, or triggering an attack by phagocytic cells, e.g., macrophages

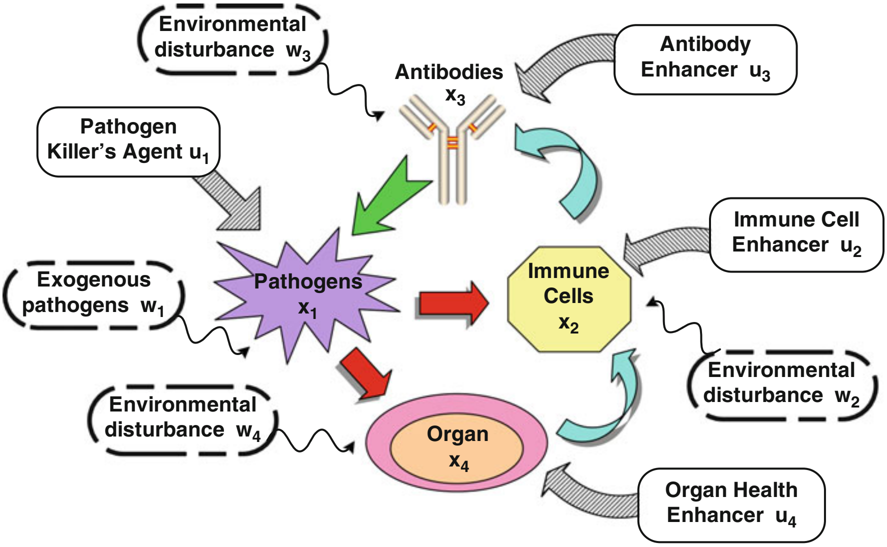

Environmental disturbance w_3
Antibodies x_3
Antibody Enhancer u_3
Pathogen Killer's Agent u_1
Immune Cell Enhancer u_2
Exogenous pathogens w_1
Pathogens x_1
Immune Cells x_2
Environmental disturbance w_2
Environmental disturbance w_4
Organ x_4
Organ Health Enhancer u_4

Fig. 27.2 Innate and enhanced immune response to a pathogenic attack under exogenous pathogens and environmental noises

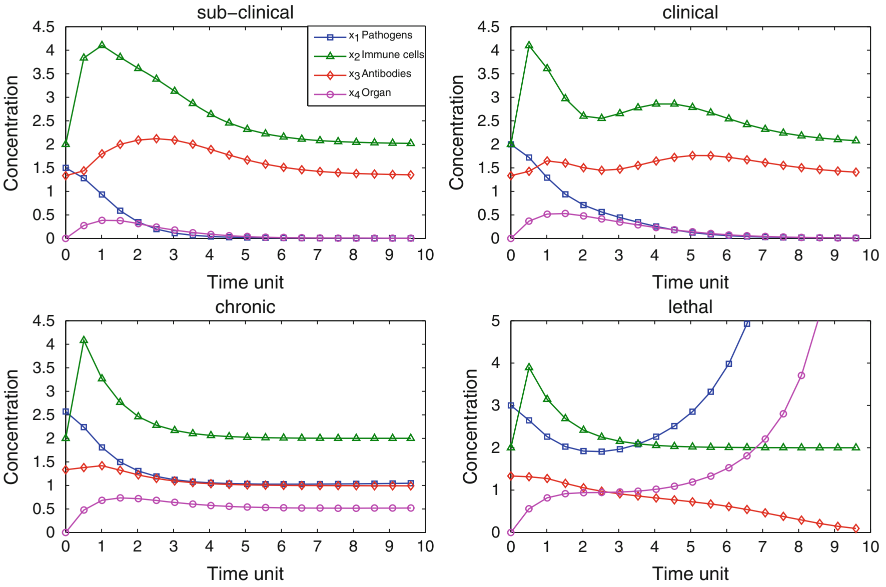


Fig. 27.3 Native immune responses to pathogens which are under sub-clinical, clinical, chronic, and lethal conditions [41]

and neutrophils. Each element of the state is subject to independent control, and new microbes may continue to enter the system.

Several typical uncontrolled responses to increasing levels of initial pathogen concentration under sub-clinical, clinical, chronic, and lethal conditions are shown in Fig. 27.3 [41]. In general, the sub-clinical response would not require medical examination, while the clinical case warrants medical consultation but is self-healing without intervention. Pathogen concentration stabilizes at non-zero values in the chronic case, which is characterized by permanently degraded organ health, and it diverges in the lethal case and expire the organ. The 'lethal' simulation of Fig. 27.3 is allowed to continue to past the point at which x_4 exceeds one [42]. Finally, a more general disease dynamic model could be represented as

$$\dot{x}(t) = f(x(t)) + Bu(t) + Dw(t), \quad x(0) = x_0 \tag{27.2}$$

where $x(t) \in \mathbf{R}^{n\times 1}$ is the state vector, $u(t) \in \mathbf{R}^{m\times 1}$ is the control agents input; $w(t) \in \mathbf{R}^{n\times 1}$ includes exogenous pathogens and environmental noises or uncertainty vector. $f(x(t))$ denotes all possible nonlinear interactions in the immune system. B denotes the matrix of control agents input coefficients (i.e., $B = diag([b_1\ b_2\ b_3\ b_4]))$. D denotes the matrix of the internal or external noises coefficients (i.e., $D = diag([d_1\ d_2\ d_3\ d_4]))$.

27.3 Robust Therapeutic Control of Immune Response

The optimal control is to specify $u(t)$ such that the following cost function is minimized within the time interval $[0, t_f]$. t_f is the final time. [41].

$$J = \frac{1}{2}E\left[x^T(t_f)Px(t_f) + \int_0^{t_f} [x^T(t)Qx(t) + u^T(t)Ru(t)]dt\right] \tag{27.3}$$

where P, Q and R are weighting matrices to be specified by designer. $E[Y]$ denotes the expectation of Y. Because the average quadratic control is only to minimize J in (27.3), i.e., the average integration of $x^T(t)Qx(t) + u^T(t)Ru(t)$ to be minimized, a control leading to large overshoot of $x(t)$ but with small integration of $x^T(t)Qx(t)$ may be specified in the quadratic optimal control design [51]. This therapeutic control will lead to organ failure because $x_4(t) \geq 1$. Furthermore, the cost function does not include exogenous pathogens and environmental noises $w(t)$, which may degrade the performance of the stochastic quadratic optimal control. Therefore, it is more appealing to prescribe a desired time response of the disease dynamic in (27.2) beforehand. Next, we design therapeutic control agents $u(t)$ to optimally track the desired time response and at the same time the influence of exogenous pathogens and environmental noises $w(t)$ on the tracking should be eliminated as much as possible.

Consider a reference model of immune system with a desired time response prescribed as follows

$$\dot{x}_r(t) = A_r x_r(t) + r(t) \tag{27.4}$$

where $x_r(t) \in \mathbf{R}^{n\times 1}$ is the reference state vector; $A_r \in \mathbf{R}^{n\times n}$ is a specific asymptotically stable matrix and $r(t)$ is a desired reference signal. It is assumed that $x_r(t)$, $\forall t > 0$ represents a desired immune response for Eq. 27.2 to follow, i.e., the therapeutic control is to specify $u(t)$ such that the tracking error $\tilde{x}(t) = x(t) - x_r(t)$ must be as small as possible under the influence of uncertain exogenous pathogens and environmental noises $w(t)$. Since the exogenous pathogens and environmental noises $w(t)$ and the initial state $x(0)$ are uncertain and reference signal $r(t)$ could be arbitrarily assigned, the robust control design should be specified so that the worst-case effect of three uncertainties $w(t)$, $x(0)$ and $r(t)$ on the tracking error could be minimized and set below a prescribed value ρ^2, i.e., both the stochastic minimax matching and robustness against uncertainties $w(t)$, $x(0)$ and $r(t)$ should be achieved simultaneously [5,9].

$$\min_{u(t)} \max_{w(t),r(t)} \frac{E\left[\int_0^{t_f} (\tilde{x}^T(t)Q\tilde{x}(t) + u^T(t)Ru(t))dt\right]}{E\left[\int_0^{t_f} (w^T(t)w(t) + r^T(t)r(t))dt + \tilde{x}^T(0)\tilde{x}(0)\right]} \leq \rho^2 \tag{27.5}$$

where the weighting matrices Q and R are assumed diagonal as follows

$$Q = diag([q_{11}\ q_{22}\ q_{33}\ q_{44}]), \quad R = diag(\left[r_{11}\ r_{22}\ r_{33}\ r_{44}\right]).$$

The diagonal element q_{ii} of Q denotes the punishment on the corresponding tracking error and the diagonal element r_{ii} of R denotes the relative therapeutic cost. Since the worst-case effect of $w(t)$, $r(t)$ and uncertain initial state $x(0)$ on tracking error $\tilde{x}(t)$ and control $u(t)$ is minimized from the energy point of view, the minimax problem of Eq. 27.5 is suitable for the stochastic minimax matching problem under unknown initial $x(0)$, uncertain environmental noises $w(t)$ and changeable reference $r(t)$, which are always met in practical design cases. Because it is not easy to solve the Nash stochastic game problem in (27.5) subject to (27.2) and (27.4) directly, we provide an upper bound ρ^2 of the minimax problem.

Remark 27.1. Actually, the design idea is the same as the model adaptive control (MRAC) [4]. The desired time response in (27.4) is the model reference in Astrom and Wittenmark. The difficulty of the model reference control design of immune system is that all the immune systems are nonlinear and external disturbances are uncertain. Therefore, the minimax game theory in (27.5) and fuzzy interpolation method are employed to simplify the design procedure of the nonlinear MRAC design problem of immune systems in the next approach.

We will first solve the above sub-minimax problem and then decrease the upper bound ρ^2 as small as possible to get the real minimax problem. Since the denominator in (27.5) is independent of $u(t)$ and is not zero, Eq. 27.5 is equivalent to [5, 9]

$$\begin{aligned}\min_{u(t)}\max_{w(t),r(t)} E\left[\int_0^{t_f}(\tilde{x}^T(t)Q\tilde{x}(t)+u^T(t)Ru(t)-\rho^2 w^T(t)w(t)-\rho^2 r^T(t)r(t))dt\right]\\ \leq \rho^2 E\left[\tilde{x}^T(0)\tilde{x}(0)\right]\end{aligned} \tag{27.6}$$

Let us denote

$$\begin{aligned}\min_{u(t)}\max_{w(t),r(t)} J(u(t),w(t),r(t)) = \min_{u(t)}\max_{w(t),r(t)} E\Bigg[\int_0^{t_f}(\tilde{x}^T(t)Q\tilde{x}(t)+u^T(t)Ru(t)\\ -\rho^2 w^T(t)w(t)-\rho^2 r^T(t)r(t))dt\Bigg]\end{aligned}$$

From the above analysis, the dynamic game problem in (27.5) or (27.6) is equivalent to finding the worst-case disturbance $w^*(t)$and reference signal $r^*(t)$ which maximize $J(u(t),w(t),r(t))$ and then a minimax control $u^*(t)$ which minimizes $J(u(t),w^*(t),r^*(t))$ such that the minimax value $J(u^*(t),w^*(t),r^*(t))$ is less than $\rho^2\tilde{x}(0)^T\tilde{x}(0)$, i.e.,

$$\begin{aligned}J(u^*(t),w^*(t),r^*(t)) &= \min_{u(t)} J(u(t),w^*(t),r^*(t))\\ &= \min_{u(t)}\max_{w(t),r(t)} J(u(t),w(t),r(t)) \leq \rho^2 E\left[\tilde{x}^T(0)\tilde{x}(0)\right]\end{aligned} \tag{27.7}$$

Hence, if there exist $u^*(t)$, $w^*(t)$, and $r^*(t)$ such that stochastic minimax matching problem in (27.7) is solved, then they can satisfy the robust performance in (27.5) as well. Therefore, the first step of robust matching control design of therapeutic agents for immune systems is to solve the following stochastic game problem.

$$\min_{u(t)} \max_{w(t),r(t)} J(u(t), w(t), r(t)) \tag{27.8}$$

subject to the disease dynamic model in (27.2) and the desired reference model in (27.4). After that, the next step is to check whether the condition $J(u^*(t), w^*(t), r^*(t)) \leq \rho^2 \tilde{x}^T(0)\tilde{x}(0)$ is satisfied or not for any $\tilde{x}(0)$.

In general, it is not easy to solve the stochastic minimax matching problem directly; it should be transformed to an equivalent minimax regulation problem. Let us denote $F(\bar{x}(t)) = \begin{bmatrix} f(x(t)) \\ A_r x_r(t) \end{bmatrix}$, $\bar{x}(t) = \begin{bmatrix} x(t) \\ x_r(t) \end{bmatrix} \in \mathbf{R}^{2n\times 1}$, $u(t) \in \mathbf{R}^{m\times 1}$ and $v(t) = \begin{bmatrix} w(t) \\ r(t) \end{bmatrix} \in \mathbf{R}^{2n\times 1}$. Then we can rewrite the stochastic minimax matching problem as

$$\begin{aligned} &\min_{u(t)} \max_{v(t)} J(u(t), v(t)) \\ &\quad = \min_{u(t)} \max_{v(t)} E\left[\int_0^{t_f} (\bar{x}^T(t)\mathbf{Q}\bar{x}(t) + u^T(t)Ru(t) - \rho^2 v^T(t)v(t))dt\right] \end{aligned} \tag{27.9}$$

subject to the following augmented system of (27.2) and (27.4)

$$\dot{\bar{x}}(t) = F(\bar{x}(t)) + \mathbf{B}u(t) + Cv(t) \tag{27.10}$$

where $\mathbf{Q} = \begin{bmatrix} Q & -Q \\ -Q & Q \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}$, $C = \begin{bmatrix} D & 0 \\ 0 & I \end{bmatrix}$, $\mathbf{I} = \begin{bmatrix} I & -I \\ -I & I \end{bmatrix}$ in which I denotes the 4-by-4 identity matrix. Then the stochastic minimax matching problem in (27.8) is equivalent to the following minimax regulation problem of the augmented system in (27.10).

$$\begin{aligned} &\min_{u(t)} \max_{v(t)} J(u(t), v(t)) \\ &\quad = \min_{u(t)} \max_{v(t)} E\left[\int_0^{t_f} (\bar{x}^T(t)\mathbf{Q}\bar{x}(t) + u^T(t)Ru(t) - \rho^2 v^T(t)v(t))dt\right] \\ &\quad \leq \rho^2 E\left[\bar{x}^T(0)\mathbf{I}\bar{x}(0)\right] \end{aligned} \tag{27.11}$$

subject to (27.10).

Theorem 27.1. *The dynamic game problem for robust matching control of immune response in (27.11) could be solved by the following stochastic minimax matching control $u^*(t)$ and the worst-case disturbance $v^*(t)$*

$$u^*(t) = -\frac{1}{2}R^{-1}\mathbf{B}^T\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)} \tag{27.12}$$

$$v^*(t) = \frac{1}{2\rho^2}C^T\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)} \tag{27.13}$$

where $V(\bar{x}(t)) > 0$ *is the positive solution of the following Hamilton-Jacobi inequality (HJI)*

$$\left(\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)^T F(\bar{x}(t)) + \bar{x}^T(t)\mathbf{Q}\bar{x}(t) - \frac{1}{4}\left(\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)^T \mathbf{B}R^{-1}\mathbf{B}^T\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}$$

$$+\frac{1}{4\rho^2}\left(\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)^T CC^T\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)} < 0 \tag{27.14}$$

with

$$V(\bar{x}(0)) \leq \rho^2\bar{x}^T(0)\mathbf{I}\bar{x}(0) \tag{27.15}$$

Proof. see Appendix A.

Since ρ^2 is the upper bound of Nash game problem in (27.5), based on the analysis above, the stochastic minimax matching control $u^*(t)$ and the worst-case disturbance $v^*(t)$ still need to minimize the upper bound ρ^2 as follows

$$\rho_0^2 = \min_{V(\bar{x}(t))>0}\rho^2 \tag{27.16}$$

subject to (27.14) and (27.15).

After solving a $V(\bar{x}(t))$ and ρ_0^2 from the constrained optimization in (27.16), we substitute this solution $V(\bar{x}(t))$ to obtain the stochastic minimax matching control $u^*(t)$ in (27.12).

27.4 Robust Control of Innate Immune System via Fuzzy Interpolation Method

Because it is very difficult to solve the nonlinear HJI in (27.14), no simple approach is available to solve the constrained optimization problem in (27.16) for robust control of innate immune system. Recently [12, 13, 44], the fuzzy T-S model has been widely applied to approximate the nonlinear system via interpolating several linearized systems at different operating points so that the nonlinear dynamic game problem could be transformed to a fuzzy dynamic game problem. Using such approach, the HJI in (27.14) can be replaced by a set of linear matrix inequalities (LMI). In this situation, the nonlinear dynamic game problem in (27.5) could be easily solved by fuzzy dynamic game method for the design of robust control for innate immune response systems.

Suppose the augmented system in (27.10) can be represented by the Takagi-Sugeno (T-S) fuzzy model [44]. The T-S fuzzy model is a piecewise interpolation of several linearized models through membership functions. The fuzzy model is described by fuzzy If-Then rules and will be employed to deal with the nonlinear dynamic game problem for robust control to achieve a desired immune response under exogenous pathogens input and environmental noises. The i-th rule of fuzzy model for nonlinear system in (27.10) is of the following form [12, 13].

Rule *i*:

If $x_1(t)$ is F_{i1} and ... and $x_g(t)$ is F_{ig}, then

$$\dot{\bar{x}}(t) = \mathbf{A}_i \bar{x}(t) + \mathbf{B}u(t) + Cv(t), \quad i = 1, 2, 3, \cdots, L \tag{27.17}$$

in which $\mathbf{A}_i = \begin{bmatrix} A_i & 0 \\ 0 & A_r \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}$, $C = \begin{bmatrix} D & 0 \\ 0 & I \end{bmatrix}$, and F_{ij} is the fuzzy set; $\mathbf{A}_i$, $\mathbf{B}$, and C are known constant matrices; L is the number of If-Then rules, g is the number of premise variables and $x_1(t), x_2(t), \ldots, x_g(t)$ are the premise variables. The fuzzy system is inferred as follows [12, 13, 44].

$$\begin{aligned}\dot{\bar{x}}(t) &= \frac{\sum_{i=1}^{L} \mu_i(x(t))[\mathbf{A}_i \bar{x}(t) + \mathbf{B}u(t) + Cv(t)]}{\sum_{i=1}^{L} \mu_i(x(t))} \\ &= \sum_{i=1}^{L} h_i(x(t))[\mathbf{A}_i \bar{x}(t) + \mathbf{B}u(t) + Cv(t)]\end{aligned} \tag{27.18}$$

where $\mu_i(x(t)) = \prod_{j=1}^{g} F_{ij}(x_j(t))$,$h_i(x(t)) = \frac{\mu_i(x(t))}{\sum_{i=1}^{L} \mu_i(x(t))}$, $x(t) = [x_1(t),$ $x_2(t), \ldots, x_g(t)]$, and $F_{ij}(x_j(t))$ is the grade of membership of $x_j(t)$ in F_{ij}.

We assume

$$\mu_i(x(t)) \geq 0 \text{ and } \sum_{i=1}^{L} \mu_i(x(t)) > 0 \tag{27.19}$$

Therefore, we get

$$h_i(x(t)) \geq 0 \text{ and } \sum_{i=1}^{L} h_i(x(t)) = 1 \tag{27.20}$$

The T-S fuzzy model in (27.18) is to interpolate L linear systems to approximate the nonlinear system in (27.10) via the fuzzy basis function $h_i(x(t))$. We specify the parameter $\mathbf{A}_i$ easily so that $\sum_{i=1}^{L} h_i(x(t))\mathbf{A}_i \bar{x}(t)$ in (27.18) can approximate $F(\bar{x}(t))$ in (27.10) by the fuzzy identification method [44].

After the nonlinear system in (27.10) is approximated as the T-S fuzzy system in (27.18), the nonlinear stochastic game problem in (27.10) and (27.11) is replaced by solving the fuzzy dynamic game problem in (27.18) and (27.11).

Theorem 27.2. *The minimax control and the worst-case disturbance for the fuzzy stochastic game problem in (27.11) subject to (27.18) are solved respectively as follows.*

$$u^*(t) = -R^{-1}\mathbf{B}^T P\bar{x}(t) \text{ and } v^*(t) = \frac{1}{\rho^2}C^T P\bar{x}(t) \tag{27.21}$$

where P is the positive definite symmetric matrix solution of the following Riccati-like inequality

$$\begin{aligned} &P\mathbf{A}_i + \mathbf{A}_i^T P + \mathbf{Q} - P^T\mathbf{B}R^{-1}\mathbf{B}^T P + \tfrac{1}{\rho^2}P^T CC^T P \le 0, \quad i = 1,\ldots,L \\ &P \le \rho^2\mathbf{I} \end{aligned} \tag{27.22}$$

Proof. see Appendix B.

By fuzzy approximation, obviously, the HJI in (27.14) can be approximated by a set of algebraic inequalities in (27.22).

Since ρ^2 is the upper bound of minimax Nash stochastic game problem in (27.5), the minimax stochastic game problem still needs to minimize ρ^2 as follows

$$\rho_0^2 = \min_{P>0} \rho^2 \tag{27.23}$$

subject to (27.22).

In order to solve the above constrained optimization in (27.23) by the conventional LMI method, we let $W = P^{-1} > 0$. Then the equation (27.22) can be equivalent to

$$\mathbf{A}_i W + W\mathbf{A}_i^T + W\mathbf{Q}W - \mathbf{B}R^{-1}\mathbf{B}^T + \frac{1}{\rho^2}CC^T \le 0, \quad i = 1,\ldots,L$$

or $\mathbf{A}_i W + W\mathbf{A}_i^T + W\begin{bmatrix} Q & -Q \\ -Q & Q \end{bmatrix} W - \mathbf{B}R^{-1}\mathbf{B}^T + \frac{1}{\rho^2}CC^T \le 0, \quad i = 1,\ldots,L$

or $\mathbf{A}_i W + W\mathbf{A}_i^T + W\begin{bmatrix} Q^{1/2} \\ -Q^{1/2} \end{bmatrix} I \begin{bmatrix} Q^{1/2} & -Q^{1/2} \end{bmatrix} W - \mathbf{B}R^{-1}\mathbf{B}^T + \frac{1}{\rho^2}CC^T \le 0$

By the Schur complements [9], the constrained optimization in (27.22) and (27.23) is equivalent to the following LMI-constrained optimization:

$$\rho_0^2 = \min_{W>0} \rho^2 \tag{27.24}$$

subject to

$$\begin{aligned} &\begin{bmatrix} \mathbf{A}_i W + W\mathbf{A}_i^T - \mathbf{B}R^{-1}\mathbf{B}^T + \frac{1}{\rho^2}CC^T & W\begin{bmatrix} Q^{1/2} \\ -Q^{1/2} \end{bmatrix} \\ \begin{bmatrix} Q^{1/2} & -Q^{1/2} \end{bmatrix} W & -I \end{bmatrix} \le 0, \quad i = 1,\ldots,L \\ &\rho^2 W \ge I \end{aligned} \tag{27.25}$$

Remark 27.2. 1. By applying fuzzy interpolation method, the nonlinear system can be approximated to several linearized systems via fuzzy basis function $h_i(x(t))$ in (27.18) and specify the constant parameter $\mathbf{A}_i$, i.e., $\sum_{i=1}^{L} h_i(x(t))\mathbf{A}_i\bar{x}(t)$. Then, the HJI in (27.14) of nonlinear stochastic game problem is replaced by a set of inequalities in (27.22), which can be easily solved by LMI-constrained optimization in (27.25).

2. The constrained optimization to solve ρ_0 and $W = P^{-1}$ in (27.24) and (27.25) can be easily solved by decreasing ρ^2 until there exists no $W > 0$ solution in (27.25). After solving W and then $P = W^{-1}$ from the constrained optimization problem in (27.24) and (27.25), the minimax control can be obtained from (27.21).
3. The solution $W > 0$ in LMI-constrained optimization (27.25) can be solved by Robust Control Toolbox in Matlab efficiently.
4. If the conventional stochastic quadratic optimal control in (27.3) is considered [42], i.e., the effect of noises is not considered in the design procedure, the optimal tracking control problem is equivalent to letting $\rho^2 = \infty$ in (27.5) [51]. Then the optimal control design $u^*(t) = -R^{-1}\mathbf{B}P\bar{x}(t)$ can be solved by a common positive definite symmetric matrix P from the equation (27.22) with $\rho^2 = \infty$, i.e., solving a common positive definite symmetric matrix $P > 0$ from the following constrained inequalities $P\mathbf{A}_i + \mathbf{A}_i^T P + \mathbf{Q} - P^T\mathbf{B}R^{-1}\mathbf{B}^T P \leq 0$, $i = 1 \cdots L$ [9]. In order to solve the optimal tracking control by LMI technique, the stochastic optimal tracking control is equivalent to solving a common $W = P^{-1}$ from the following constrained inequalities

$$\mathbf{A}_i W + W\mathbf{A}_i^T + W\begin{bmatrix} Q^{1/2} \\ -Q^{1/2} \end{bmatrix} I \begin{bmatrix} Q^{1/2} & -Q^{1/2} \end{bmatrix} W - \mathbf{B}R^{-1}\mathbf{B}^T \leq 0, \quad i = 1, \ldots, L \tag{27.26}$$

or equivalently,

$$\begin{bmatrix} \mathbf{A}_i W + W\mathbf{A}_i^T - \mathbf{B}R^{-1}\mathbf{B}^T & W\begin{bmatrix} Q^{1/2} \\ -Q^{1/2} \end{bmatrix} \\ \begin{bmatrix} Q^{1/2} & -Q^{1/2} \end{bmatrix} W & -I \end{bmatrix} \leq 0, \quad i = 1, \ldots, L \tag{27.27}$$

which is equivalent to (27.25) with $\rho^2 = \infty$.

According to the analysis above, the robust control of innate immune system via fuzzy interpolation method is summarized as follows.

Design Procedure:

1. Give a desired reference model in (27.4) of immune system.
2. Select membership functions and construct fuzzy plant rules in (27.17).
3. Give weighting matrices Q and R in (27.5).

4. Solve the LMI-constrained optimization in (27.25) to obtain W (thus $P = W^{-1}$ can also be obtained) and ρ_0^2.
5. Construct the controller under the worst-case noises in (27.21).

27.5 Computational Simulation

We consider the innate immune system in (27.1) and in Fig. 27.2. The values of the parameters are in the Table 27.1. The immune noises $w_1 \sim w_4$ are assumed white Gaussian noises. One is the external noises which are under infectious situation; the microbes infect the organ not only by an initial concentration at the beginning but also by the continuous pathogens input. For the convenience of computer simulation, suppose the continuous pathogens input to the immune system is viewed as an environmental disturbance w_1. The other is internal noises $w_2 \sim w_4$ are assumed zero mean white noises with standard deviations all equal to 2. The dynamic model of innate immune system under exogenous pathogens input and environmental noises are controlled by a combined therapeutic control shown in (27.1) [41] with the set of the initial condition $x(0) = [x_1(0)x_2(0)x_3(0)x_4(0)]^T = [33.110.98]^T$. In this example, therapeutic controls $u_1 \sim u_4$ are combined to enhance the immune system.

Our reference model design objective is that system matrix A_r and $r(t)$ should be specified beforehand so that its transient responses and steady state of reference system for innate immune response system are desired. If the real parts of eigenvalues of A_r are more negative (i.e., more robust stable), the tracking system will be more robust to environmental noises. After some numerical simulations for clinical treatment, the desired reference signals are obtained by the following

Table 27.1 Model parameters of dynamic innate immune system [24,42]

Parameter	Value	Description
a_{11}	1	Pathogens reproduction rate coefficient
a_{12}	1	The suppression by pathogens coefficient
a_{22}	3	Immune reactivity coefficient
a_{23}	1	The mean immune cell production rate coefficient
x_2^*	2	The steady-state concentration of immune cells
a_{31}	1	Antibodies production rate coefficient
a_{32}	1.5	The antibody mortality coefficient
a_{33}	0.5	The rate of antibodies suppress pathogens
a_{41}	0.5	The organ damage depends on the pathogens damage possibilities coefficient
a_{42}	1	Organ recovery rate
b_1	-1	Pathogen killer's agent coefficient
b_2	1	Immune cell enhancer coefficient
b_3	1	Antibody enhancer coefficient
b_4	-1	Organ health enhancer coefficient
$d_1 \sim d_4$	3	Noises coefficient

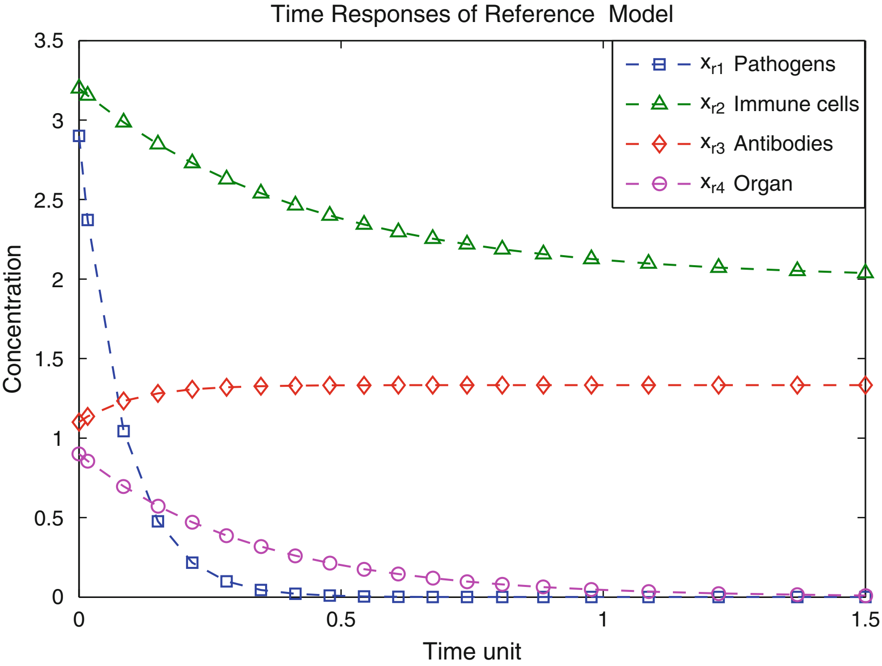


Fig. 27.4 The desired reference model with four desired states in (27.28): pathogens (x_{r1}, *blue, dashed square line*), immune cells (x_{r2}, *green, dashed triangle line*), antibodies (x_{r3}, *red, dashed diamond line*) and organ (x_{r4}, *magenta, dashed circle line*)

reference model (see Fig. 27.4).

$$\dot{x}_r(t) = A_r \cdot x_r(t) + B_r \cdot u_{step}(t) \tag{27.28}$$

where $A_r = diag(\left[-12\ -2.3\ -10\ -3\right])$ $\quad B_r = \left[0\ 4.6\ 13.3333\ 0\right]^T$ and $u_{step}(t)$ is the unit step function.

From the investigation of the uncontrolled innate immune response (lethal case) in Fig. 27.5, the pathogen concentration is increasing rapidly and causes organ failure. We try to administrate a treatment after a period of pathogens infection to enhance the immune system. The cutting line (black solid line) in Fig. 27.5 is a proper time to take drugs. Suppose the set of the initial condition of the desired reference model is about $x_r(0) = \left[2.9\ 3.2\ 1.1\ 0.9\right]^T$. The time response of the desired reference model in (27.28) is shown in Fig.27.4.

To minimize the design effort and complexity for this nonlinear innate immune system in (27.1), we employ the T-S fuzzy model to construct fuzzy rules to approximate the nonlinear innate immune system with the innate immune system's state variables as premise variables in the following.

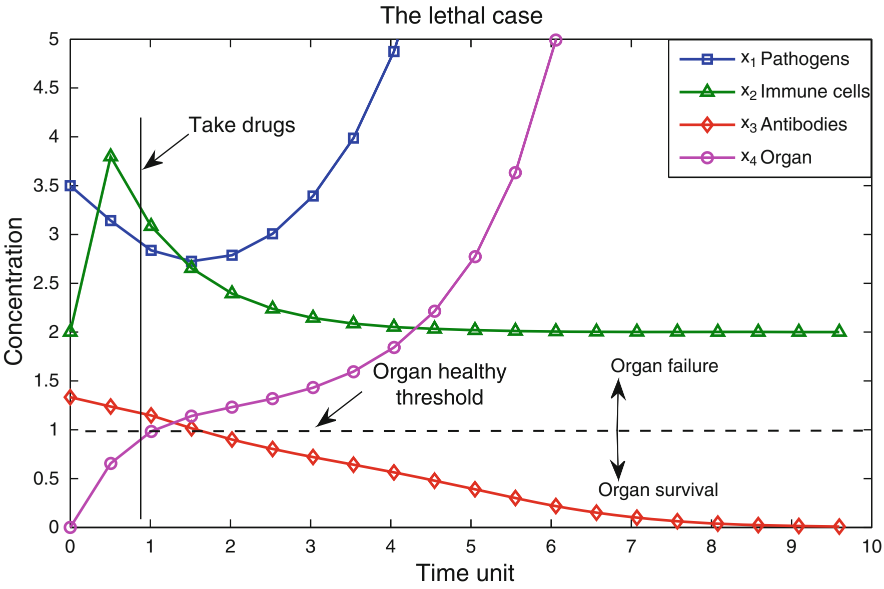


Fig. 27.5 The uncontrolled immune responses (lethal case) in (27.1) are shown to increase the level of pathogen concentration at the beginning of the time period. In this case, we try to administrate a treatment after a short period of pathogens infection. The cutting line (*black solid line*) is an optimal time point to give drugs. The organ will survive or fail based on the organ health threshold (*horizontal dashed line*) [x_4 <1: survival, x_4 ≥1: failure]

Rule *i*:

If x_1 is F_{i1}, x_2 is F_{i2}, x_3 is F_{i3}, and x_4 is F_{i4}, then

$$\dot{\bar{x}}(t) = \mathbf{A}_i \bar{x}(t) + \mathbf{B}u(t) + Cv(t), \quad i = 1, 2, 3, \cdots, L \tag{27.29}$$

where $\bar{x} = [x_1\ x_2\ x_3\ x_4\ x_{r1}\ x_{r2}\ x_{r3}\ x_{r4}]^T$, $u = [u_1\ u_2\ u_3\ u_4]^T$, $v = [w_1\ w_2\ w_3\ w_4\ r_1\ r_2\ r_3\ r_4]^T$, the number of the fuzzy rules is $L = 16$. To construct the fuzzy model in (27.29), we need to find the operating points of the innate immune response. Suppose the operating points for x_1 are at $\bar{x}_{11} = 0$, and $\bar{x}_{12} = 4$. Similarly, the operating points of x_2, x_3, and x_4 are at $\bar{x}_{21} = 0$, $\bar{x}_{22} = 10$, $\bar{x}_{31} = 0$, $\bar{x}_{32} = 5$, $\bar{x}_{41} = 0$, and $\bar{x}_{42} = 1$, respectively. For the convenience of design, triangle type membership functions are taken for Rule 1 through Rule 16. We create two triangle type membership functions for each state at these operating points (see Fig. 27.6). In order to accomplish the robust matching performance, we should tune up a set of the weighting matrices Q and R of the cost function in (27.11) as follows

$$Q = diag(\begin{bmatrix} 1\ 1\ 1\ 1 \end{bmatrix}), \quad R = diag(\begin{bmatrix} 0.003\ 0.003\ 0.003\ 0.003 \end{bmatrix}).$$

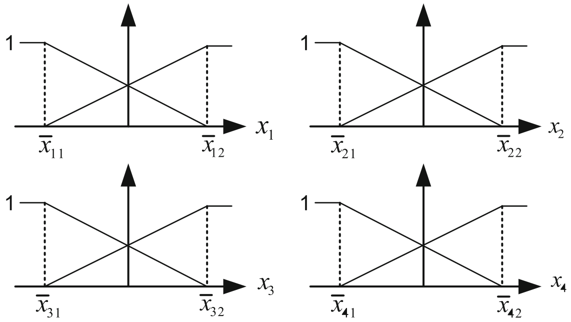


Fig. 27.6 Membership functions for four states x_1, x_2, x_3 and x_4

After specifying the desired reference model, we need to solve the constrained optimization in (27.24) for the robust minimax control in (27.21) by employing Matlab Robust Control Toolbox. Finally, we obtain a minimum noise attenuation level $\rho_0^2 = 0.98$ and a common positive definite symmetric matrix P for Eq. 27.22 as follows

$$P = \begin{bmatrix} 0.43313 & 0 & 0 & 0 & -0.43313 & 0 & 0 & 0 \\ 0 & 0.56172 & 0 & 0 & 0 & -0.56172 & 0 & 0 \\ 0 & 0 & 0.42678 & 0 & 0 & 0 & -0.42678 & 0 \\ 0 & 0 & 0 & 0.28482 & 0 & 0 & 0 & -0.28482 \\ -0.43313 & 0 & 0 & 0 & 0.50151 & 0 & 0 & 0 \\ 0 & -0.56172 & 0 & 0 & 0 & 0.62738 & 0 & 0 \\ 0 & 0 & -0.42678 & 0 & 0 & 0 & 0.49526 & 0 \\ 0 & 0 & 0 & -0.28482 & 0 & 0 & 0 & 0.34567 \end{bmatrix}$$

Figures 27.7 and 27.8 present the simulation results for the robust control. Figure 27.7 shows the responses of the controlled immune system by minimax model matching control with the concentrations of the pathogens x_1, immune cells x_2, antibodies x_3 and organ index x_4 to track the desired reference states x_{r1}, x_{r2}, x_{r3} and x_{r4}, respectively. From the simulation results, the tracking performance of the robust control via T-S fuzzy interpolation is quite satisfactory. The Fig. 27.8 shows the four combined therapeutic control signals. Obviously, from Figs. 27.7 and 27.8, it is seen that the effect of stochastic external noise on the reference model tracking of immune system is attenuated significantly by the proposed robust therapeutic control design.

27.6 Discussion

From the simulation results (Figs. 27.7 and 27.8), it is shown that the innate immune system under the continuous intrusion of exogenous pathogens and the corruption of environmental noises can be controlled by a robust control design to achieve the

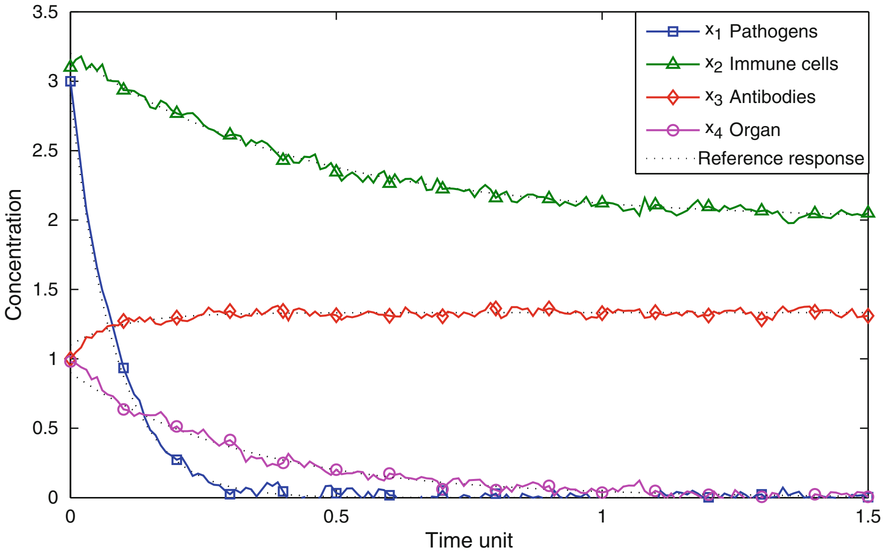


Fig. 27.7 The tracking of innate immune system to the desired reference model by the robust stochastic minimax matching control under the continuous exogenous pathogens and environmental noises

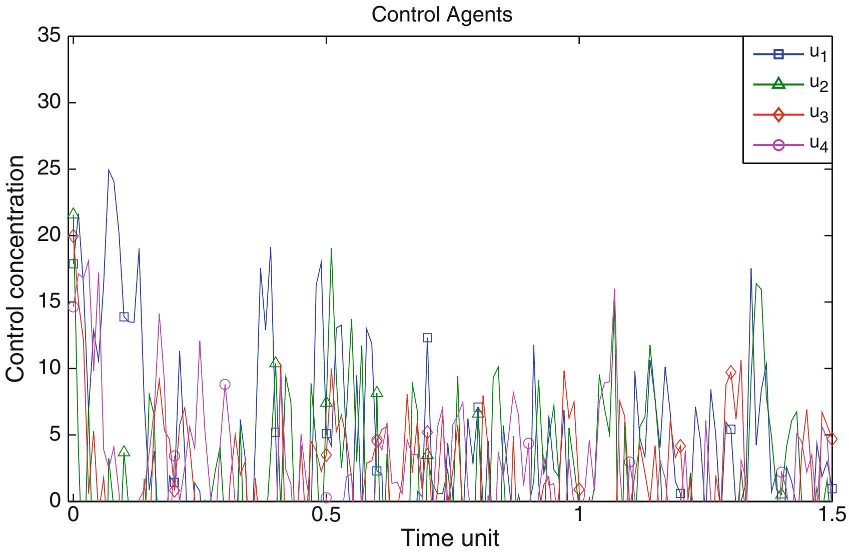


Fig. 27.8 The stochastic minimax controls in the simulation example. The drug controls u_1 (*blue, solid square line*) for pathogens, u_2 for immune cells (*green, solid triangle line*), u_3 for antibodies (*red, solid diamond line*) and u_4 for organ (*magenta, solid circle line*)

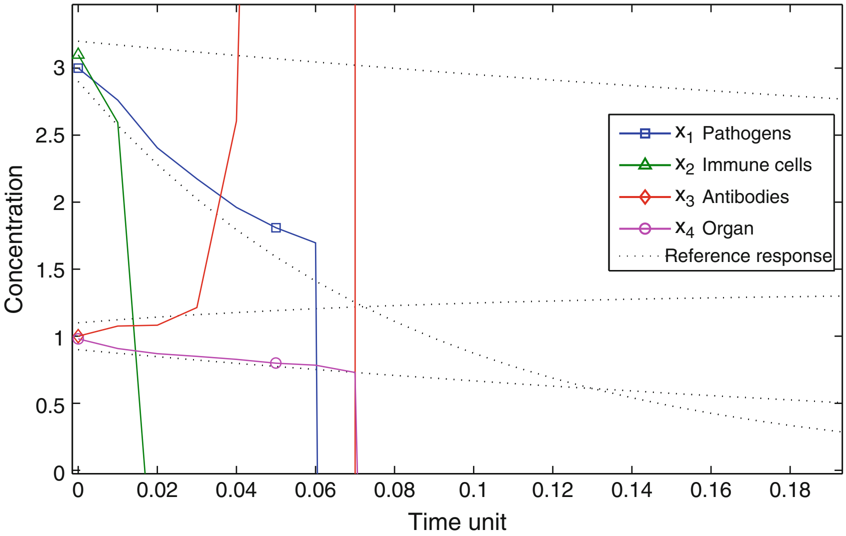


Fig. 27.9 In the case of conventional stochastic optimal control (i.e., $\rho = \infty$ in (27.5)), since the effect of continuous exogenous pathogens intrusion and environmental noises is not considered in the design procedure, the states of innate immune system overshoot and diverge and cannot track the desired reference responses

desired time response. If we consider the conventional optimal control in (27.3), i.e., the effect of the environmental noises is not included in the cost function; the optimal tracking control problem is equivalent to letting $\rho^2 = \infty$ in (27.5) and (27.22). From the simulation results (Figs. 27.9 and 27.10), the four states of optimal tracking of the immune system are overshooting and diverging without tracking the desired immune time response. Obviously, exogenous pathogens and the environmental noises have deteriorated the optimal tracking performance and therefore their effects should be considered in the robust control design procedure. In the situation, the proposed robust matching control design is necessary to achieve a desired time response.

The combined therapies design is an important issue for all human diseases [46]. For a long period, the treatment of inflammatory skin diseases such as psoriasis, contact dermatitis and atopic dermatitis has included agents that alleviate symptoms, but these agents have not been aimed at any specific molecular targets involved in the pathogenesis of the disease. Insights into this immune mechanism may facilitate the development of combination therapies that take advantage of the robust design, with the aim of achieving higher efficacy at a lower drug dosage. The proposed robust design has used four control variables, i.e., pathogen killer's agent u_1, immune cell enhancer u_2, antibody enhancer u_3, and health enhancer u_4, to achieve a stochastic minimax matching performance and to efficiently attenuate the effect of exogenous pathogens and environmental noises on the immune system.

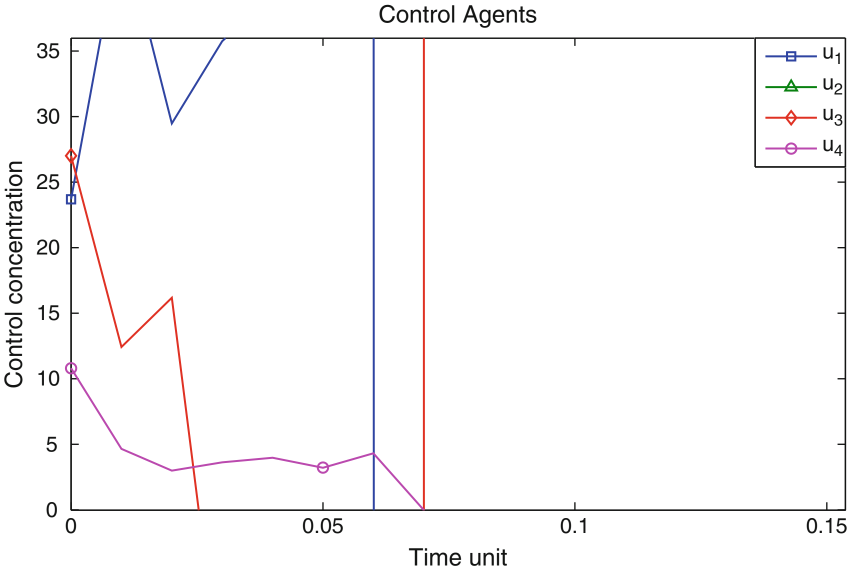


Fig. 27.10 The controls of conventional stochastic optimal control (i.e., $\rho = \infty$ in (27.5)) without considering the effect of continuous exogenous pathogens intrusion and environmental noises in the design procedure, the drug controls u_1 (*blue, solid square line*) for pathogens, u_2 for immune cells (*green, solid triangle line*), u_3 for antibodies (*red, solid diamond line*) and u_4 for organ (*magenta, solid circle line*) are all overshooting and divergent

In this study, the model of innate immune dynamic system is taken from the literature, which still needs to compare quantitatively with empirical evidence in practical application. For practical implementation, accurate biodynamic models are required for treatment application. However, model identification is not the topic of this paper. Furthermore, we have made an assumption that the four states $(x_1 \sim x_4)$ of the concentrations or indices can be measured accurately by the medical equipment. With these detectable signals, we can solve these stochastic game problems for robust tracking control design of innate immune system to obtain the drug administration values in real time through medical instrument readout. If measurement is corrupted by noises in the measurement process, some filter designs should be employed to attenuate these noises to estimate the state variables for control in (27.21) [40]. Nevertheless, the implementation of filter will increase the complexity of the design problem [2]. Since the proposed robust control design can provide an efficient way to create a real time therapeutic regime to protect suspected patients from the pathogens infection, in the future, we will focus on applications of robust control design to therapy and drug design incorporating with nanotechnology and metabolic engineering scheme.

As a comparison, the similarity and difference between our stochastic minimax control method and the stochastic optimal control method [38] are given in the following. We all want to minimize the total cost of design, i.e., the weighted sum of the damage caused by pathogens and the cost paid by specific immune cells. On the other hand, the differences are given in the following: (1) A desired model reference is given to be optimally tracked for the enhancement of the immune system in our proposed method and Shudo and Iwasa have designed an optimal control to minimize the cost. (2) The effect of external noises has not been considered in Shudo and Iwasa, but the worst-case effect of external noises has been considered and minimized in our design via the Nash stochastic game method. (3) Fuzzy interpolation technique is employed by our method so that linear matrix inequalities (LMIs) technique is used to efficiently solve the nonlinear minimax optimization problem in our design procedure. However, Shudo and Iwasa have used a dynamic programming method to derive an optimal schedule to solve the nonlinear optimization problem of host defense, immune memory and post-infection pathogen levels in mammals.

27.7 Conclusion

Robustness is a significant property that allows the innate immune system to maintain its function despite exogenous pathogens, environmental noises (external noises) and system uncertainties (internal noises). Based on stochastic game theory, the robust tracking control is formulated as a stochastic minimax problem for an innate immune system to achieve a desired time response prescribed prior under environmental noises, unknown initial conditions. In general, the robust control design for innate immune system needs to solve nonlinear Hamilton-Jacobi inequality (HJI), which is generally difficult to solve for this control design. Based on the proposed fuzzy stochastic game scheme, the design of nonlinear dynamic robust matching control problem for innate immune system is transformed to solve a set of equivalent linear stochastic game problem. Such transformation can then allow us an easier approach by solving a LMI-constrained optimization problem for robust minimax control design. With the help of the Robust Control Toolbox in Matlab instead of the HJI, we could solve these linear stochastic game problems for robust matching control of innate immune system efficiently. From the *in silico* simulation examples, the proposed stochastic minimax match control of immune system could track the prescribed reference time response robustly, which may lead to potential application in therapeutic drug design for a desired immune response during an infection episode.

Appendix

Appendix A

Proof of Theorem 1.

Let us denote a Lyapunov energy function $V(\bar{x}(t)) > 0$. Then (27.9) is equivalent to the following minimax problem

$$\min_{u(t)}\max_{v(t)} J = \min_{u(t)}\max_{v(t)} E\left[V(\bar{x}(0)) - V(\bar{x}(t_f)) + \int_0^{t_f}\left(\bar{x}^T(t)\mathbf{Q}\bar{x}(t) + u^T(t)Ru(t) - \rho^2 v^T(t)v(t) + \frac{dV(\bar{x}(t))}{dt}\right)dt\right] \tag{A1}$$

By the chain rule, we get

$$\frac{dV(\bar{x}(t))}{dt} = \left(\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)^T \cdot \frac{d\bar{x}(t)}{dt} = \left(\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)^T \cdot (F(\bar{x}(t)) + \mathbf{B}u(t) + Cv(t)) \tag{A2}$$

Substituting (A2) into (A1), we get

$$\begin{aligned}\min_{u(t)}\max_{v(t)} J = \min_{u(t)}\max_{v(t)} E\Bigg[&V(\bar{x}(0)) - V(\bar{x}(t_f)) + \int_0^{t_f} (\bar{x}^T(t)\mathbf{Q}\bar{x}(t) \\ &+ u^T(t)Ru(t) - \rho^2 v^T(t)v(t) + \left(\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)^T F(\bar{x}(t)) \\ &+ \left(\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)^T \mathbf{B}u(t) + \left(\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)^T Cv(t))dt\Bigg]\end{aligned}$$

$$\begin{aligned}\min_{u(t)}\max_{v(t)} J = \min_{u(t)}\max_{v(t)} E\Bigg[&V(\bar{x}(0)) - V(\bar{x}(t_f)) + \int_0^{t_f} \Bigg(\left(\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)^T F(\bar{x}(t)) \\ &+ \bar{x}(t)^T\mathbf{Q}\bar{x}(t) - \frac{1}{4}\left(\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)^T \mathbf{B}R^{-1}\mathbf{B}^T\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)} \\ &+ \frac{1}{4\rho^2}\left(\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)^T CC^T\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)} \\ &+ \left(u^T(t) + \frac{1}{2}R^{-1}\mathbf{B}^T\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)^T R\left(u(t) + \frac{1}{2}R^{-1}\mathbf{B}^T\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right) \\ &- \left(\rho v(t) - \frac{1}{2\rho}C^T\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)^T\left(\rho v(t) - \frac{1}{2\rho}C^T\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)\Bigg)dt\Bigg]\end{aligned}$$

Therefore, the minimax solution is given as follows

$$\begin{aligned}J(u^*(t), v^*(t)) = E\Bigg[&V(\bar{x}(0)) - V(\bar{x}(t_f)) + \int_0^{t_f}\Bigg\{\left(\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)^T F(\bar{x}(t)) \\ &+ \bar{x}(t)^T\mathbf{Q}\bar{x}(t) - \frac{1}{4}\left(\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)^T \mathbf{B}R^{-1}\mathbf{B}^T\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)} \\ &+ \frac{1}{4\rho^2}\left(\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\right)^T CC^T\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}\Bigg\}dt\Bigg]\end{aligned}$$

with

$$u^*(t) = -\frac{1}{2}R^{-1}\mathbf{B}^T\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}$$
$$v^*(t) = \frac{1}{2\rho^2}C^T\frac{\partial V(\bar{x}(t))}{\partial \bar{x}(t)}$$

if the equation (27.14) holds, then

$$J(u^*(t), v^*(t)) \leq E\left[V(\bar{x}(0))\right] - E\left[V(\bar{x}(t_f))\right]$$

From the inequality in (27.11), this minimax solution should be less than $\rho^2\bar{x}^T(0)\mathbf{I}\bar{x}(0)$, and then we get the inequality in (27.15).

$$\min_{u(t)}\max_{v(t)} J \leq E\left[V(\bar{x}(0))\right] - E\left[V(\bar{x}(t_f))\right] \leq E\left[V(\bar{x}(0))\right] \leq \rho^2 E\left[\bar{x}^T(0)\mathbf{I}\bar{x}(0)\right]$$

Appendix B

Proof of Theorem 2.

Let us denote a Lyapunov energy function $V(\bar{x}(t)) = \bar{x}^T(t)P\bar{x}(t) > 0$. Then (27.9) is equivalent to the following

$$\begin{aligned}
&\min_{u(t)}\max_{v(t)} J \\
&\quad = \min_{u(t)}\max_{v(t)} E\Big[\bar{x}^T(0)P\bar{x}(0) - \bar{x}^T(t_f)P\bar{x}(t_f) \\
&\qquad + \int_0^{t_f}\left(\bar{x}^T(t)\mathbf{Q}\bar{x}(t) + u^T(t)Ru(t) - \rho^2 v^T(t)v(t) + \frac{dV(\bar{x}(t))}{dt}\right)dt\Big] \\
&\quad = \min_{u(t)}\max_{v(t)} E\Big[\bar{x}^T(0)P\bar{x}(0) - \bar{x}^T(t_f)P\bar{x}(t_f) \\
&\qquad + \int_0^{t_f}\left(\bar{x}^T(t)\mathbf{Q}\bar{x}(t) + u^T(t)Ru(t) - \rho^2 v^T(t)v(t) + 2\bar{x}^T(t)P\dot{\bar{x}}(t)\right)dt\Big] \\
&\quad = \min_{u(t)}\max_{v(t)} E\Big[\bar{x}^T(0)P\bar{x}(0) - \bar{x}^T(t_f)P\bar{x}(t_f) \\
&\qquad + \int_0^{t_f}\Big(\bar{x}^T(t)\mathbf{Q}\bar{x}(t) + u^T(t)Ru(t) - \rho^2 v^T(t)v(t) \\
&\qquad + 2\bar{x}^T(t)P\left(\sum_{i=1}^{L} h_i(x(t))\mathbf{A}_i\bar{x}(t)\right) + 2\bar{x}^T(t)P\mathbf{B}u(t) + 2\bar{x}^T(t)PCv(t)\Big)dt\Big]
\end{aligned}$$

$$
\begin{aligned}
&= \min_{u(t)}\max_{v(t)} E\Bigg[\bar{x}^T(0)P\bar{x}(0) - \bar{x}^T(t_f)P\bar{x}(t_f)\\
&\quad + \int_0^{t_f}\Big(\bar{x}^T(t)\mathbf{Q}\bar{x}(t) + u^T(t)Ru(t) - \rho^2 v^T(t)v(t)\\
&\quad + \sum_{i=1}^{L} h_i(x(t))\Big(2\bar{x}^T(t)P\mathbf{A}_i\bar{x}(t) + 2\bar{x}^T(t)P\mathbf{B}u(t) + 2\bar{x}^T(t)PCv(t)\Big)\Big)dt\Bigg]\\
&= \min_{u(t)}\max_{v(t)} E\Bigg[\bar{x}^T(0)P\bar{x}(0) - \bar{x}^T(t_f)P\bar{x}(t_f) + \int_0^{t_f}\Big\{\bar{x}^T(t)\mathbf{Q}\bar{x}(t)\\
&\quad + \sum_{i=1}^{L} h_i(x(t))\bar{x}^T(t)\Big[P\mathbf{A}_i + \mathbf{A}_i^T P - P^T\mathbf{B}R^{-1}\mathbf{B}^T P + \frac{1}{\rho^2}P^T CC^T P\Big]\bar{x}(t)\\
&\quad + \Big(u(t) + R^{-1}\mathbf{B}^T P\bar{x}(t)\Big)^T R\Big(u(t) + R^{-1}\mathbf{B}^T P\bar{x}(t)\Big)\\
&\quad - \Big(\rho v(t) - \frac{1}{\rho}C^T P\bar{x}(t)\Big)^T\Big(\rho v(t) - \frac{1}{\rho}C^T P\bar{x}(t)\Big)\Big\}\,dt\Bigg]
\end{aligned}
$$

The minimax solution is given as follows

$$
\begin{aligned}
J(u^*(t), v^*(t)) = E\Bigg[&\bar{x}^T(0)P\bar{x}(0) - \bar{x}^T(t_f)P\bar{x}(t_f) + \int_0^{t_f}\Big\{\bar{x}^T(t)\mathbf{Q}\bar{x}(t)\\
&+ \sum_{i=1}^{L} h_i(x(t))\bar{x}^T(t)\Big[P\mathbf{A}_i + \mathbf{A}_i^T P - P^T\mathbf{B}R^{-1}\mathbf{B}^T P\\
&+ \frac{1}{\rho^2}P^T CC^T P\Big]\bar{x}(t)\Big\}\,dt\Bigg]
\end{aligned}
$$

with $u^*(t) = -R^{-1}\mathbf{B}^T P\bar{x}(t)$ and $v^*(t) = \frac{1}{\rho^2}C^T P\bar{x}(t)$.

In order to simplify the above equation, suppose the inequality in (27.22) holds, then

$$\min_{u(t)}\max_{v(t)} J \le E\left[\bar{x}(0)P\bar{x}(0)\right]$$

From the inequality in (27.11), this minimax should be less than $\rho^2 E\left[\bar{x}^T(0)\mathbf{I}\bar{x}(0)\right]$, and then

$$\min_{u(t)}\max_{v(t)} J \le E\left[\bar{x}(0)P\bar{x}(0)\right] \le \rho^2 E\left[\bar{x}^T(0)\mathbf{I}\bar{x}(0)\right], \text{i.e., } P \le \rho^2\mathbf{I}$$

Since we assume ρ^2 is the upper bound in (27.5), the minimax control becomes how to design $u^*(t)$ in (27.21) by solving the constrained optimization problem in (27.22) and (27.23).

References

1. Adler, F. R., & Karban, R. (1994). Defended fortresses or moving targets? Another model of inducible defenses inspired by military metaphors. *American Naturalist*, *144*, 813–832.
2. Althaus, C. L., Ganusov, V. V., & De Boer, R. J. (2007). Dynamics of CD8+ T cell responses during acute and chronic lymphocytic choriomeningitis virus infection. *Journal of Immunology*, *179*, 2944–2951.
3. Asachenkov, A. L. (1994). *Disease dynamics*. Boston: Birkhuser.
4. Astrom, K. J., & Wittenmark, B. (1995). *Adaptive control*. Reading, Mass: Addison-Wesley.
5. Basar, T., & Olsder, G. J. (1999). *Dynamic noncooperative game theory*. Philadelphia: SIAM.
6. Bell, D. J., & Katusiime, F. (1980). A Time-Optimal Drug Displacement Problem. *Optimal Control Applications & Methods*, *1*, 217–225.
7. Bellman, R. (1983). *Mathematical methods in medicine*. Singapore: World Scientific.
8. Bonhoeffer, S., May, R. M., Shaw, G. M., & Nowak, M. A. (1997). Virus dynamics and drug therapy. *Proceedings of the National Academy Sciences of the United States of America*, *94*, 6971–6976.
9. Boyd, S. P. (1994). Linear matrix inequalities in system and control theory. *Society for Industrial and Applied Mathematics*, Philadelphia.
10. Carson, E. R., Cramp, D. G., Finkelstein, F., & Ingram, D. (1985). Computers and control in clinical medicine. In *Control system concepts and approaches in clinical medicine*. New York: Plenum.
11. Chen, B. S., Chang, C. H., & Chuang, Y. J. (2008). Robust model matching control of immune systems under environmental disturbances: Dynamic game approach. *Journal of Theoretical Biology*, *253*, 824–837.
12. Chen, B. S., Tseng, C. S., & Uang, H. J. (1999). Robustness design of nonlinear dynamic systems via fuzzy linear control. *IEEE Transactions on Fuzzy Systems*, *7*, 571–585.
13. Chen, B. S., Tseng, C. S., & Uang, H. J. (2000). Mixed H-2/H-infinity fuzzy output feedback control design for nonlinear dynamic systems: An LMI approach. *IEEE Transactions on Fuzzy Systems*, *8*, 249–265.
14. Chizeck, H., & Katona, P. (1985). Computers and control in clinical medicine. In *Closed-loop control* (pp. 95–151). New York: Plenum.
15. De Boer, R. J., & Boucher, C. A. (1996). Anti-CD4 therapy for AIDS suggested by mathematical models. *Proceedings of Biological Science*, *263*, 899–905.
16. Gentilini, A., Morari, M., Bieniok, C., Wymann, R., & Schnider, T. W. (2001). Closed-loop control of analgesia in humans. In *Proceedings of the IEEE conference on decision and control* (Vol. 1, pp. 861–866). Orlando.
17. Janeway, C. (2005). *Immunobiology: The immune system in health and disease*. New York: Garland.
18. Jelliffe, R. W. (1986). Topics in clinical pharmacology and therapeutics. In *Clinical applications of pharmacokinetics and control theory: Planning, monitoring, and adjusting regimens of aminoglycosides, lidocaine, digitoxin, and digoxin* (pp. 26–82). New York: Springer.
19. Kirschner, D., Lenhart, S., & Serbin, S. (1997). Optimal control of the chemotherapy of HIV. *Journal of Mathematical Biology*, *35*, 775–792.
20. Kwong, G. K., Kwok, K. E., Finegan, B. A., & Shah, S. L. (1995). Clinical evaluation of long range adaptive control for meanarterial blood pressure regulation. In *Proceedings of the American control conference* (Vol. 1, pp. 786–790). Seattle.
21. Li, T. H. S., Chang, S. J., & Tong, W. (2004). Fuzzy target tracking control of autonomous mobile robots by using infrared sensors. *IEEE Transactions on Fuzzy Systems*, *12*, 491–501.
22. Lian, K. Y., Chiu, C. S., Chiang, T. S., & Liu, P. (2001). LMI-based fuzzy chaotic synchronization and communications. *IEEE Transactions on Fuzzy Systems*, *9*, 539–553.
23. Lydyard, P. M., Whelan, A., & Fanger, M. W. (2000). *Instant notes in immunology*. New York: Springer.
24. Marchuk, G. I. (1983). Mathematical models in immunology. In *Optimization software. Inc. Worldwide distribution rights*. New York: Springer.

25. Nowak, M. A., & May, R. M. (2000). *Virus dynamics : Mathematical principles of immunology and virology*. Oxford: Oxford University Press.
26. Nowak, M. A., May, R. M., Phillips, R. E., Rowland-Jones, S., Lalloo, D. G., McAdam, S., Klenerman, P., Koppe, B., Sigmund, K., Bangham, C. R., et al. (1995). Antigenic oscillations and shifting immunodominance in HIV-1 infections. *Nature*, *375*, 606–611.
27. Parker, R. S., Doyle, J. F., III, Harting, J. E., & Peppas, N. A. (1996). Model predictive control for infusion pump insulin delivery. In *Proceedings of the 18th annual international conference of the IEEE engineering in medicine and biology society* (Vol. 5, pp. 1822–1823). Amsterdam.
28. Perelson, A. S., Kirschner, D. E., & De Boer, R. (1993). Dynamics of HIV infection of CD4+ T cells. *Mathematical Bioscience*, *114*, 81–125.
29. Perelson, A. S., Neumann, A. U., Markowitz, M., Leonard, J. M., & Ho, D. D. (1996). HIV-1 dynamics in vivo: virion clearance rate, infected cell life-span, and viral generation time. *Science*, *271*, 1582–1586.
30. Perelson, A. S., & Weisbuch, G. (1997). Immunology for physicists. *Reviews of Modern Physics*, *69*, 1219–1267.
31. Polycarpou, M. M., & Conway, J. Y. (1995). Modeling and control of drug delivery systems using adaptive neural control methods. In *Proceedings of the American control conference* (Vol. 1, pp. 781–785). Seattle.
32. Robinson, D. C. (1986). Topics in clinical pharmacology and therapeutics. In *Principles of pharmacokinetics* (pp. 1–12). New York: Springer.
33. Rundell, A., HogenEsch, H., & DeCarlo, R. (1995). Enhanced modeling of the immune system to incorporate naturalkiller cells and memory. In *Proceedings of American control conference* (Vol. 1, pp. 255–259). Seattle.
34. Schumitzky, A. (1986). Topics in clinical pharmacology and therapeutics. In *Stochastic control of pharmacokinetic systems* (pp. 13–25). New York: Springer.
35. Shudo, E., Haccou, P., & Iwasa, Y. (2003). Optimal choice between feedforward and feedback control in gene expression to cope with unpredictable danger. *Journal of Theoretical Biology*, *223*, 149–160.
36. Shudo, E., & Iwasa, Y. (2001). Inducible defense against pathogens and parasites: optimal choice among multiple options. *Journal of Theoretical Biology*, *209*, 233–247.
37. Shudo, E., & Iwasa, Y. (2002). Optimal defense strategy: Storage vs. new production. *Journal of Theoretical Biology*, *219*, 309–323.
38. Shudo, E., & Iwasa, Y. (2004). Dynamic optimization of host defense, immune memory, and post-infection pathogen levels in mammals. *Journal of Theoretical Biology*, *228*, 17–29.
39. Stafford, M. A., Corey, L., Cao, Y., Daar, E. S., Ho, D. D., & Perelson, A. S. (2000). Modeling plasma virus concentration during primary HIV infection. *Journal of Theoretical Biology*, *2003*, 285–301.
40. Stengel, R. F., & Ghigliazza, R. (2004). Stochastic optimal therapy for enhanced immune response. *Mathematical Bioscience*, *191*, 123–142.
41. Stengel, R. F., Ghigliazza, R., Kulkarni, N., & Laplace, O. (2002). Optimal control of innate immune response. *Optimal Control Applications & Methods*, *23*, 91–104.
42. Stengel, R. F., Ghigliazza, R. M., & Kulkarni, N. V. (2002). Optimal enhancement of immune response. *Bioinformatics*, *18*, 1227–1235.
43. Swan, G. W. (1981). Optimal-Control Applications in Biomedical-Engineering – a Survey. *Optimal Control Applications & Methods*, *2*, 311–334.
44. Takagi, T., & Sugeno, M. (1985). Fuzzy Identification of Systems and Its Applications to Modeling and Control. *IEEE Transactions on Systems Man and Cybernetics*, *15*, 116–132.
45. van Rossum, J. M., Steyger, O., van Uem, T., Binkhorst, G. J., & Maes, R. A. A. (1986). Modelling of biomedical systems. In *Pharmacokinetics by using mathematical systems dynamics* (pp. 121–126). Amsterdam: Elsevier.
46. Villadsen, L. S., Skov, L., & Baadsgaard, O. (2003). Biological response modifiers and their potential use in the treatment of inflammatory skin diseases. *Experimental Dermatology*, *12*, 1–10.

47. Wein, L. M., D'Amato, R. M., & Perelson, A. S. (1998). Mathematical analysis of antiretroviral therapy aimed at HIV-1 eradication or maintenance of low viral loads. *Journal of Theoretical Biology*, *192*, 81–98.
48. Wiener, N. (1948). *Cybernetics; or, control and communication in the animal and the machine*. Cambridge: Technology Press.
49. Wodarz, D., & Nowak, M. A. (1999). Specific therapy regimes could lead to long-term immunological control of HIV. *Proceedings of the National Academy of Sciences of the United States of America*, *96*, 14464–14469.
50. Wodarz, D., & Nowak, M. A. (2000). CD8 memory, immunodominance, and antigenic escape. *European Journal of Immunology*, *30*, 2704–2712.
51. Zhou, K., Doyle, J. C., & Glover, K. (1996). *Robust and optimal control*. Upper Saddle River, NJ: Prentice Hall.