LNBI 6282

Tjeerd M.H. Dijkstra
Evgeni Tsivtsivadze
Elena Marchiori
Tom Heskes (Eds.)

Pattern Recognition in Bioinformatics

5th IAPR International Conference, PRIB 2010
Nijmegen, The Netherlands, September 2010
Proceedings

IAPR

Springer

Lecture Notes in Bioinformatics 6282

Edited by S. Istrail, P. Pevzner, and M. Waterman

Subseries of Lecture Notes in Computer Science

Tjeerd M.H. Dijkstra   Evgeni Tsivtsivadze
Elena Marchiori   Tom Heskes (Eds.)

Pattern Recognition in Bioinformatics

5th IAPR International Conference, PRIB 2010
Nijmegen, The Netherlands, September 22-24, 2010
Proceedings

Springer

Series Editors

Sorin Istrail, Brown University, Providence, RI, USA
Pavel Pevzner, University of California, San Diego, CA, USA
Michael Waterman, University of Southern California, Los Angeles, CA, USA

Volume Editors

Tjeerd M.H. Dijkstra
Evgeni Tsivtsivadze
Elena Marchiori
Tom Heskes
Radboud University Nijmegen, Institute for Computing and Information Sciences
Heyendaalseweg 135, 6525AJ Nijmegen, The Netherlands
E-mail: {t.dijkstra, evgeni, t.heskes}@science.ru.nl; elenam@cs.ru.nl

Library of Congress Control Number: 2010934523

CR Subject Classification (1998): J.3, I.5, F.2.2, I.2, H.3.3, H.2.8

LNCS Sublibrary: SL 8 – Bioinformatics

ISSN	0302-9743
ISBN-10	3-642-16000-X Springer Berlin Heidelberg New York
ISBN-13	978-3-642-16000-4 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

The field of bioinformatics has two main objectives: the creation and maintenance of biological databases and the analysis of life sciences data in order to unravel the mysteries of biological function. Computer science methods such as pattern recognition, machine learning, and data mining have a great deal to offer the field of bioinformatics. The Pattern Recognition in Bioinformatics (PRIB) meeting was established in 2006 under the auspices of the International Association of Pattern Recognition (IAPR) to create a focus for the application and development of computer science methods to life science data.

The 5th PRIB conference was held in Nijmegen, The Netherlands, on 22–24 September 2010. A total of 46 papers were submitted to the conference for peer review. Of those, 38 (83%) were accepted for publication in these proceedings. The invited speakers were Rita Casadio (Bologna Biocomputing Group, Italy), Florence d'Alché-Buc (Université d'Evry-Val d'Essonne, France), Daniel Huson (Tübingen University, Germany), and Natasa Przulj (Imperial College London, UK). Tutorials were delivered by Concettina Guerra (Università di Padova, Italy), Clarisse Dhaenens (Laboratoire LIFL/INRIA, France), Laetitia Jourdan (Laboratoire LIFL/INRIA, France), Neil Lawrence (University of Manchester, UK), and Dick de Ridder (Delft University of Technology).

We would like to thank all authors who spent time and effort to contribute to this book and the members of the Program Committee for their evaluation of the submitted papers. We are grateful to Nicole Messink for her administrative help and coordination, to the co-organizers of this conference, to the machine learning group members for their assistance before and during the conference, and to the EasyChair team (http://easychair.org) for providing the conference review management system. We acknowledge support from the Netherlands Organization for Scientific Research (NWO), the Netherlands Bioinformatics Centre (NBIC), the Radboud University Nijmegen, the SIKS Netherlands research School for Information and Knowledge Systems, the royal dutch science society (KNAW) and the EU FP7 network of excellence Pascal2.

Finally, we hope that you will consider contributing to PRIB 2011.

July 2010

Tjeerd MH Dijkstra
Evgeni Tsivtsivadze
Tom Heskes
Elena Marchiori

Organization

Conference Chairs

Elena Marchiori	Radboud University Nijmegen
Tom Heskes	Radboud University Nijmegen

Steering Committee

Jagath Rajapakse	Nanyang Technological University
Ray Acharya	Pennsylvania State University
Guido Sanguinetti	University of Sheffield
Madhu Chetty	Monash University
Visakan Kadirkamanathan	University of Sheffield

Program Chairs

Tjeerd Dijkstra	Radboud University Nijmegen
Guido Sanguinetti	University of Sheffield
Visakan Kadirkamanathan	University of Sheffield

Special Sessions Chair

Lutgarde Buydens	Radboud University Nijmegen

Publicity Chair

Jin-Kao Hao	University of Angers

Tutorial Chair

Bert Kappen	Radboud University Nijmegen

Webmaster

Evgeni Tsivtsivadze	Radboud University Nijmegen

Program Committee

Jesus Aguilar, Spain
Shandar Ahmad, Japan
Florence d'Alché-Buc, France
Tatsuya Akutsu, Japan
Jaume Bacardit, UK
Karsten Borgwardt, Germany
Rainer Breitling, The Netherlands
Nicolas Brunel, France
Sebastian Böcker, Germany
William Bush, USA
C.Q. Chang, Hong Kong
Frédéric Cazals, France
Marco Chierici, Italy
Colin Campbell, UK
Theo Damoulas, UK
Federico Divina, Spain
Bas Dutilh, The Netherlands
Richard Edwards, UK
Alexandru Floares, Romania
Maurizio Filippone, UK
Cesare Furlanello, Italy
Raul Giraldez, Spain
Rosalba Giugno, Italy
Michael Habeck, Germany
Jennifer Hallinan, UK
Jin-Kao Hao, France
Morihiro Hayashida, Japan
Tom Heskes, The Netherlands
Antti Honkela, Finland
Pavol Jancura, The Netherlands
Zhenyu Jia, USA
Rasa Jurgelenaite, The Netherlands
Giuseppe Jurman, Italy
Visakan Kadirkamanathan, UK
Seyoung Kim, USA
Walter Kosters, The Netherlands
Mehmet Koyuturk, USA
Krishna Murthy Karuturi, Singapore
Guillaume Launay, France
Kee Khoon Lee, Singapore
Pietro Lio', UK
Xuejun Liu, China
Stefano Lonardi, USA
Elena Marchiori, The Netherlands
Francesco Masulli, Italy
Vadim Mottl, Russia
Jason Moore, USA
Alison Motsinger-Reif, USA
Sach Mukherjee, UK
Tamas Nepusz, UK
Mahesan Niranjan, UK
Josselin Noirel, UK
Richard Notebaart, The Netherlands
Carlotta Orsenigo, Italy
Alberto Paccanaro, UK
Andrea Passerini, Italy
Thang Pham, The Netherlands
Clara Pizzuti, Italy
Esa Pitkänen, Finland
Beatriz Pontes, Spain
Marylyn Ritchie, USA
Simon Rogers, UK
Juho Rousu, Finland
Miguel Rocha, Portugal
Gunnar Rätsch, Germany
Yvan Saeys, Belgium
Guido Sanguinetti, UK
Jun Sese, Japan
Evangelos Simeonidis, UK
Jennifer Smith, USA
Kieran Smallbone, UK
Johan Suykens, Belgium
Roberto Tagliaferri, Italy
Alexey Tsymbal, Germany
Jing Yang, China
Haixuan Yang, UK
Hong Yan, Hong Kong
Andrew Zammit, UK

Table of Contents

Part I: Classification of Biological Sequences

Part II: Unsupervised Learning Methods for Biological Sequences

Part III: Learning Methods for Gene Expression and Mass Spectrometry Data

Part IV: Bioimaging

Part V: Molecular Structure Prediction

Part VI: Protein Protein Interaction and Network Inference

Part I

Classification of Biological Sequences

Sequence-Based Prediction of Protein Secretion Success in *Aspergillus niger*

Bastiaan A. van den Berg[1,2,4], Jurgen F. Nijkamp[1,4], Marcel J.T. Reinders[1,2,4], Liang Wu[3], Herman J. Pel[3], Johannes A. Roubos[3], and Dick de Ridder[1,2,4]

[1] The Delft Bioinformatics Lab, Delft University of Technology, The Netherlands
[2] Netherlands Bioinformatics Centre, The Netherlands
[3] DSM Biotechnology Center, The Netherlands
[4] Kluyver Centre for Genomics of Industrial Fermentation, The Netherlands
b.a.vandenberg@tudelft.nl

Abstract. The cell-factory *Aspergillus niger* is widely used for industrial enzyme production. To select potential proteins for large-scale production, we developed a sequence-based classifier that predicts if an over-expressed homologous protein will successfully be produced and secreted. A dataset of 638 proteins was used to train and validate a classifier, using a 10-fold cross-validation protocol. Using a linear discriminant classifier, an average accuracy of 0.85 was achieved. Feature selection results indicate what features are mostly defining for successful protein production, which could be an interesting lead to couple sequence characteristics to biological processes involved in protein production and secretion.

Keywords: *Aspergillus niger*, protein secretion, sequence-based prediction, classification.

1 Introduction

The filamentous fungus *Aspergillus niger* has a high secretion capacity, which makes it an ideal cell-factory widely used for industrial production of enzymes [11]. Selecting proteins for large-scale production requires testing for successful over-expression and protein secretion. Because many proteins are of potential interest, a large amount of lab work is needed. This can be reduced by developing a software tool to prioritize proteins in advance. Such a tool might also indicate which gene or protein characteristics influence successful over-expression and secretion.

Various sequence-based classifiers have been developed, for example, to predict protein crystallization propensity [6], protein solubility [8], and protein subcellular localization [14], [4]. Subcellular localization predictors have been used to predict protein secretion [16], [5], but these methods predict if a protein is inherently extracellular, whereas our aim is to predict *successful* secretion of a protein after over-expression.

In this work, we present a classifier to predict if a homologous protein will successfully be secreted after over-expression in *A. niger*, using 25 sequence-based features and providing an accuracy of 0.85.

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 3–14, 2010.

2 Materials and Methods

2.1 Data Set

The data set D contained 638 homologous proteins from *A. niger* CBS 513.88 [13] with a signal sequence predicted by SignalP [12]. For each protein, the open reading frame (ORF) and a binary score for successful over-expression was given. To obtain this binary success score, each protein was over-expressed through introduction of the predicted gene using the same strong glucoamylase promoter P_{GlaA}. Cultures were grown in shake-flasks and the filtered broth was put on an SDS-PAGE gel. Successful over-expression was defined as the detection of a visible band in this gel. D contained 268 successfully detected proteins (D_{pos}), and 370 unsuccessfully detected proteins (D_{neg}). The data set will be publicly available soon.

2.2 Sequence-Based Features

For each item $i \in D$, a feature vector $\boldsymbol{d_i}$ with 39 sequence-based features was constructed (Table 1). Next to simple compositional features, features that relate to protein solubility and membrane binding were chosen, because it is expected that these characteristics influence successful protein secretion. Features are calculated using the entire ORF sequence and corresponding protein sequence, including the signal peptide. A two-sample t-test with pooled variance estimation was used as class separability criterion to evaluate the performance of each feature. Features with p-value > 0.001 (gray features in Table 1) were removed, resulting in a set of 25 features used for classifier development.

For this set of features, a heat map of the hierarchical clustered (complete linkage) feature matrix is shown in Fig. 1, in which each row is a protein in D and each column is a feature. The two additional columns on the right depict the measured and predicted class labels. They show that clustering of the proteins, using this feature set, already provides a separation of D_{pos} and D_{neg}.

Compositional Features. Given a protein sequence, its amino acid composition is defined as the number of occurrences of the amino acid (frequency count) divided by the sequence length, providing 20 features. The same was done for the nucleotide composition of the coding region, providing 4 features.

Additionally, we calculated the compositions of amino acid sets that share a common property. Given a protein sequence and an amino acid set, the amino acid set composition is defined as the sum of the frequency counts of each of the specified amino acids, divided by the sequence length. Eight sets were used: helix $\{I,L,F,W,Y,V\}$, turn $\{N,G,P,S\}$, sheet $\{A,E,L,M\}$, charged $\{R,D,C,E,H,K,Y\}$, small $\{A,N,D,C,G,P,S,T,V\}$, tiny $\{A,G,S\}$, basic $\{R,K,H\}$, and acidic $\{N,D,E,Q\}$. One nucleotide set was used: GC.

As final compositional feature we used the codon adaptation index (CAI)[15], which was calculated with the codon usage index of all genes in the *A. niger* genome.

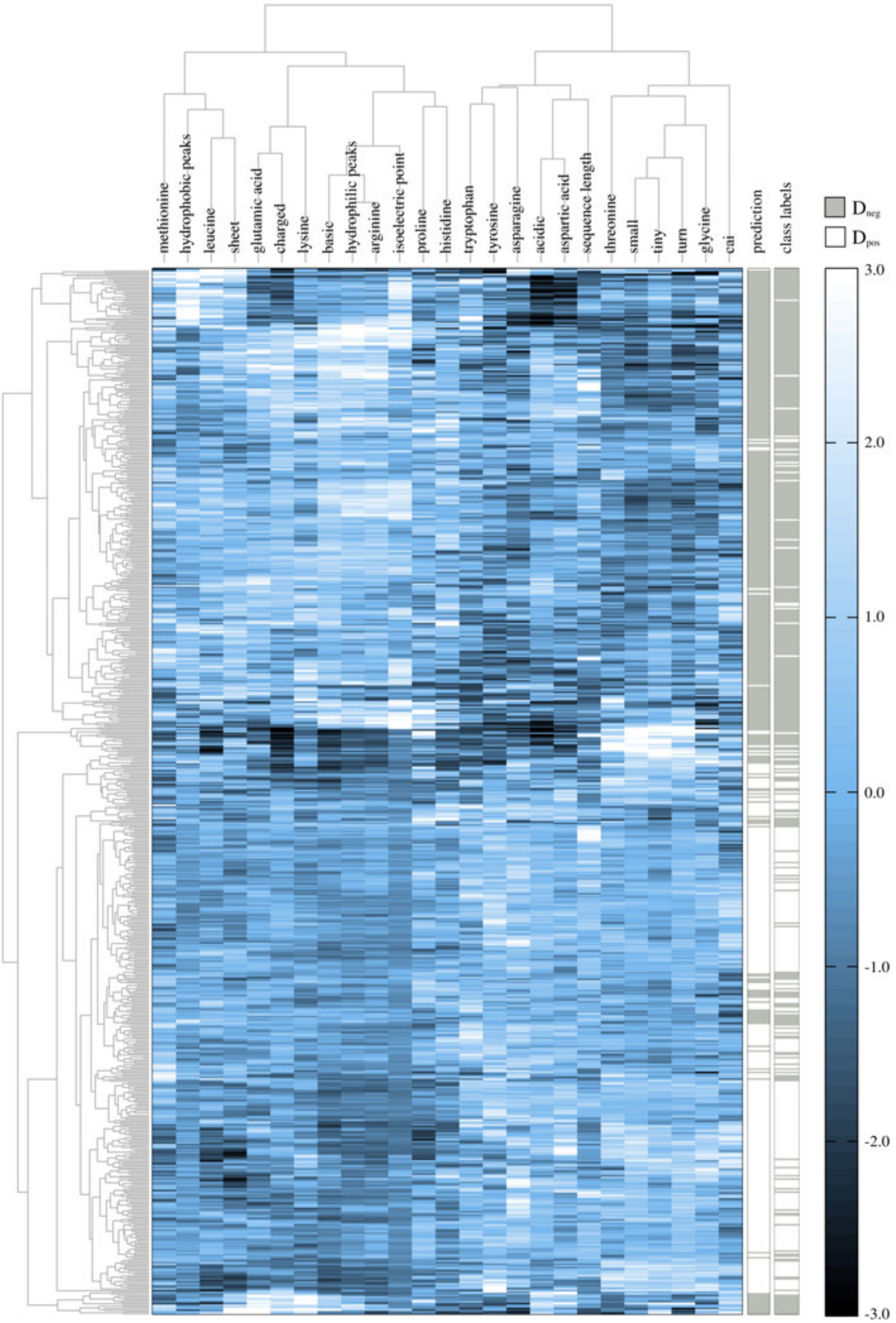


Fig. 1. Heat map of clustered feature matrix. The rows are the proteins in D and the columns are the 25 features used for classifier development. The two columns on the right depict the predicted and measured class labels respectively.

Table 1. Calculated features with class separability score

Nucleotide compositional	guanine (2.5)	GC (1.3)	
	adenine (0.4)	CAI (5.3)	
	thymine (2.3)		
	cytosine (2.9)		
Amino acid compositional	alanine (2.3)	leucine (9.0)	helix {I,L,F,W,Y,V} (0.4)
	arginine (13.6)	lysine (9.3)	turn {N,G,P,S} (8.9)
	asparagine (15.0)	methionine (6.3)	sheet {A,E,L,M} (10.8)
	aspartic acid (7.2)	phenylalanine (0.1)	acidic {N,D,E,Q} (7.9)
	cysteine (0.2)	proline (5.4)	basic {R,K,H} (15.7)
	glutamic acid (5.6)	serine (1.6)	charged {R,D,C,E,H,K,Y} (5.6)
	glutamine (0.2)	threonine (8.3)	small {A,N,D,C,G,P,S,T,V} (9.7)
	glycine (9.2)	tryptophan (6.3)	tiny {A,G,S} (3.5)
	histidine (4.2)	tyrosine (13.6)	
	isoleucine (0.9)	valine (1.9)	
Signal-based features	hydrophobic peaks (9.1)		
	hydrophilic peaks (15.5)		
Global features	GRAVY (1.8)		
	isoelectric point (16.2)		
	sequence length (5.4)		

Signal-based Features. Two features capture the occurrence of local hydropathic peaks: *hydrophobic peaks* and *hydrophilic peaks*, both derived from a protein hydropathicity signal [1] that was constructed using the (normalized) hydropathicity amino acid scale of Kyte and Doolitle [7].

An *amino acid scale* is defined as a mapping from each amino acid to a value. Given a protein sequence, a hydropathicity signal was obtained by replacing each residue by its amino acid scale value (Fig. 2A). The signal was smoothed through convolution with a triangular function (Fig. 2B). To capture the extreme values of the smoothed signal, an upper and lower threshold were set (Fig. 2C). *Hydrophobic peaks* is defined as the sum of all areas above the upper threshold divided by the sequence length, *hydrophilic peaks* is defined as the sum of all areas below the lower threshold divided by the sequence length.

The window size and edge of the triangular function (Fig. 2B), and both thresholds (Fig. 2C) can be varied. In each CV loop of the training and validation protocol (Section 2.4), an exhaustive search was applied to optimize the features' class separability score, using: $window\ size = 3, 5, \ldots, 21$; $edge = 0.0, 0.2, \ldots, 1.0$; $threshold = 0.5, 0.54, \ldots, 0.86$ for *hydrophobic peaks* and 0.5, 0.45, ..., 0.05 for *hydrophilic peaks*.

Global Features. Three global features were used: the grand average of hydrophobicity (GRAVY), i.e., the sum of all Kyte and Doolitle amino acid scale values divided by the sequence length; the isoelectric point (pI), i.e., the predicted pH at which the net charge of the protein is zero; and finally the sequence length, i.e., the number of residues in the protein sequence.

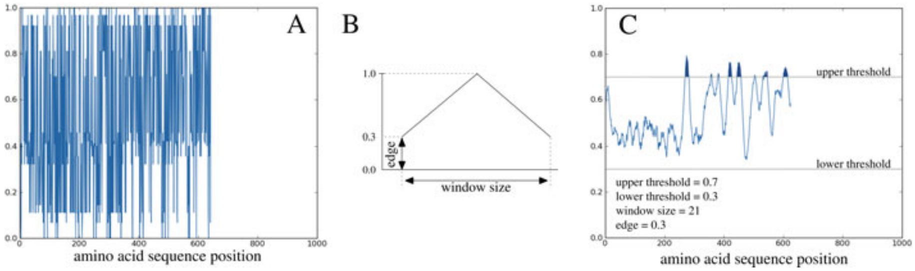


Fig. 2. Hydropathic peaks features. A) A raw protein hydropathicity signal obtained by replacing each amino acid in the sequence by its value in the normalized Kyte and Doolitle amino acid scale. **B)** Triangular function used to smooth the raw signal. **C)** Smoothed signal obtained by convolution of the raw signal in A with the function in B.

WoLF PSORT. To test whether using predicted localization would improve performance, WoLF PSORT [4] was used to predict secretion of the proteins in D. Next to the amino acid composition and the sequence length, which we also used as features, WoLF PSORT uses features based on sorting signals and functional motifs. To use the prediction as feature, we assigned proteins with intracellular localization prediction a value of 0, and proteins predicted to be extracellular a value of 1.

2.3 Performance Evaluation

We used five measures to evaluate classification performance. Four of these are based on the confusion matrix. This matrix contains the number of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Let the set of positives be $P = TP + FN$, the set of negatives $N = TN + FP$, the set of predicted positives $P' = TP + FP$, and the set of predicted negatives $N' = TN + FN$. The confusion matrix-based measures are; $accuracy = (TP + TN)/(P + N)$, $sensitivity = TP/P$, $specificity = TN/N$, and Matthews correlation coefficient score $MCC = (TP \times TN - FP \times FN)/\sqrt{P \times N \times P' \times N'}$. The MCC-score [9] is suited in case of different class sizes, which applies in our case. The score ranges from 0 for random assignment, to 1 for perfect prediction.

The aforementioned scores take into account only one operating point on the receiver operating characteristic (ROC) curve. As a fifth measure, we took the area under the ROC curve (AUC), thereby taking into account a range of operating points. Because the goal is to reduce the amount of lab work, we are mainly interested in low false positive rates, i.e., the left region of the ROC-curve. Therefore, we used the AUC over the range of $0 - 0.3$ false positive rate (ROC0.3) as main performance measure.

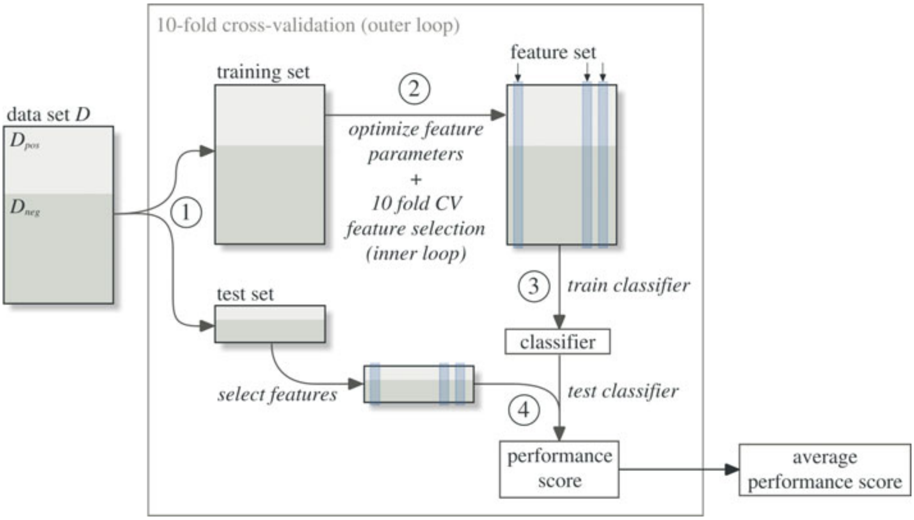


Fig. 3. Training and validation protocol

2.4 Training and Validation Protocol

To avoid overestimation of classification performance, a double 10-fold CV protocol was used, based on the protocol in [17]. We used 10-fold CV feature selection with classifier performance as selection criterion, in which the expected error $((FP/P + FN/N)/2)$ was used as performance measure.

The protocol is shown in Fig. 3. The dataset D is split into ten equal-sized random stratified sets. In each outer loop, one of the sets is used as test set, and the remaining nine as the training set (1). An exhaustive search is done to optimize the parameters of the hydropathic peaks features for maximal class separability, and 10-fold CV feature selection (inner loop) is applied on the training set to select an optimal feature set (2). As feature selection methods, we used both forward and backward feature selection. The optimal feature set is used to train a classifier on the entire training set (3). The resulting classifier is applied to the test set that was not employed for training, resulting in a performance score (4). Finally, the performance scores of the 10 CV loops are averaged, resulting in an average performance score.

The training and validation protocol was implemented in Matlab, using the PRTools pattern recognition toolbox [3].

2.5 Classifiers

We tested 8 classifiers: linear and quadratic normal density-based Bayes classifiers (ldc, qdc); nearest mean classifier (nmc); k-nearest neighbor classifier, both with $k = 1$ and with k optimized by leave-one-out CV (1nnc, knnc), naive Bayes classifier (naivebc), Fisher's least square linear classifier (fisherc), and a radial basis support vector machine (svm, $\gamma = 1/\text{number of features}$). We used libsvm [2] for the support vector machine.

3 Results

The classifier performance scores are given in Table 2. We compared the ROC0.3 scores of the different methods using a paired t-test ($p < 0.05$) on the results of the 10 CV loops. This showed that the nearest neighbor classifiers perform significantly worse than all other methods, except for qdc with forward feature selection. The best performance was obtained with ldc and backward feature selection.

Table 2. Classifier performance scores

classifier		ROC0.3	sensitivity	specificity	MCC	accuracy
ldc	f[1]	0.232 ±0.03	0.877 ±0.08	0.819 ±0.06	0.691 ±0.08	0.843 ±0.04
	b[2]	**0.236** ±0.03	0.873 ±0.08	0.830 ±0.05	0.700 ±0.07	0.848 ±0.03
svm	f	0.228 ±0.03	0.847 ±0.08	**0.857** ±0.02	**0.701** ±0.07	**0.853** ±0.03
	b	0.232 ±0.02	0.843 ±0.08	0.854 ±0.04	0.695 ±0.09	0.850 ±0.04
fisherc	f	0.234 ±0.03	0.873 ±0.08	0.819 ±0.06	0.688 ±0.08	0.842 ±0.04
	b	0.235 ±0.02	0.881 ±0.09	0.822 ±0.05	0.698 ±0.07	0.846 ±0.03
naivebc	f	0.224 ±0.03	0.854 ±0.08	0.800 ±0.05	0.649 ±0.09	0.823 ±0.04
	b	0.230 ±0.03	0.888 ±0.08	0.803 ±0.03	0.684 ±0.07	0.839 ±0.03
qdc	f	0.221 ±0.03	0.877 ±0.06	0.803 ±0.04	0.674 ±0.06	0.834 ±0.03
	b	0.227 ±0.03	0.884 ±0.05	0.805 ±0.04	0.682 ±0.08	0.838 ±0.04
nmc	f	0.227 ±0.03	**0.910** ±0.07	0.773 ±0.04	0.678 ±0.06	0.831 ±0.02
	b	0.224 ±0.02	0.899 ±0.07	0.773 ±0.04	0.666 ±0.05	0.826 ±0.02
knnc	f	0.218 ±0.03	0.858 ±0.09	0.770 ±0.06	0.624 ±0.10	0.807 ±0.05
	b	0.214 ±0.02	0.862 ±0.06	0.778 ±0.06	0.635 ±0.05	0.813 ±0.03
1nnc	f	0.195 ±0.04	0.798 ±0.09	0.781 ±0.09	0.578 ±0.15	0.788 ±0.07
	b	0.190 ±0.03	0.809 ±0.09	0.749 ±0.08	0.557 ±0.10	0.774 ±0.05

[1] forward feature selection, [2] backward feature selection

Fig. 4 shows the ROC0.3 scores of ldcs trained on each of the 25 single features, on all 25 features, and on features obtained by backward feature selection. The classifiers are ordered by score. A paired t-test ($p < 0.001$) on the 10 CV loops showed that all single-feature classifiers are significantly outperformed by both multi-feature classifiers. Although using all features provides a higher average score than using backward feature selection, the paired t-test ($p < 0.05$) indicates that the difference is not significant.

Applying WoLF PSORT on our dataset provided a sensitivity of 0.96 and a specificity of 0.49. It appears that WoLF PSORT is too optimistic, providing a large amount of FPs. This could be explained by the difference in the problems we address; WoLF PSORT predicts extracellular proteins, whereas our method also includes successful protein production and secretion. This means that extracellular proteins in D, which are positives for WoLF PSORT, can be part

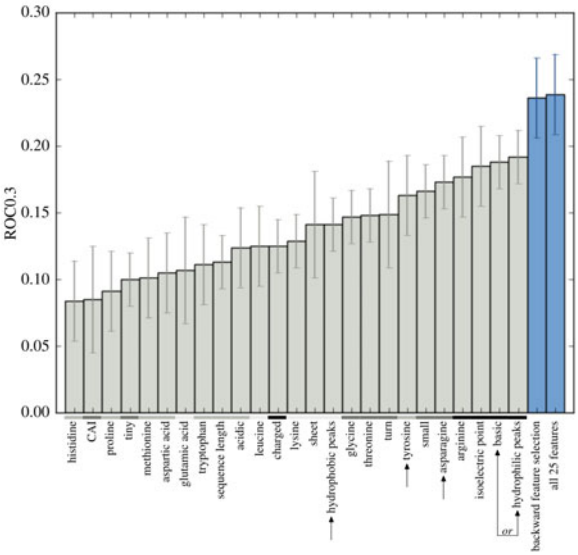


Fig. 4. Single-feature and multi-feature classification scores

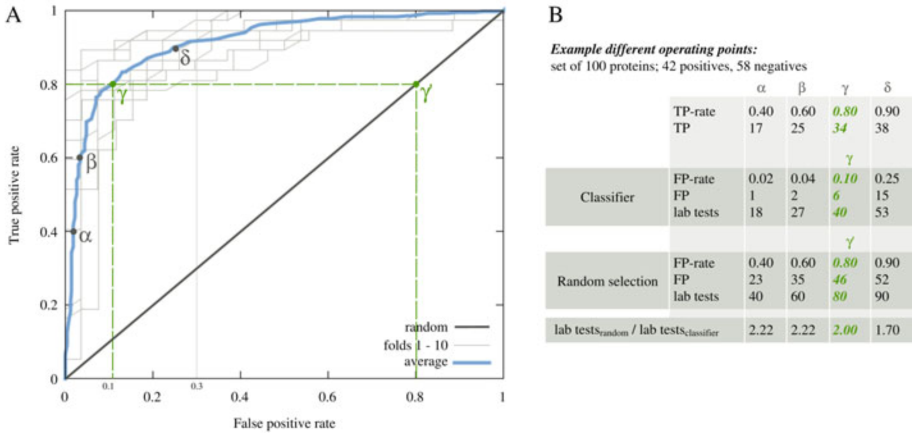
of D_{neg} because of unsuccessful protein production. We used the localization prediction as additional feature. Using ldc with backward feature selection, no significant improvement was observed, probably because the feature contains redundant data.

3.1 Operating Point Example

Fig. 5A shows the ROC of the ldc with backward feature selection. One could use this classifier to screen a set of proteins for potential over-expression candidates. For example, if we have a set S of 100 proteins that we want to screen, containing 42 positives (S_{pos}) and 58 negatives (S_{neg}) (i.e., the same fraction of positives and negatives as D), and if we use γ as operating point, a true positive rate of 0.8 will be obtained. In this case, the classifier will predict 34 true positives and 6 false positives, which means that only 40 lab experiments are needed to identify 34 positives. Without the classifier, to identify 34 positives, both the false and the true positive rate will be 0.8 (operating point γ'). In this case, 80 lab experiments will be needed to identify 34 positives, which means that the classifier could reduce the amount of lab work by a factor two (Fig. 5B).

3.2 Feature Optimization

Fig. 6 shows the optimal parameter settings for the hydrophilic and hydrophobic peaks feature as obtained in one of the CV loops. For both features, the same optimum was observed in each CV loop.



Example different operating points:
set of 100 proteins; 42 positives, 58 negatives

		α	β	γ	δ
	TP-rate TP	0.40 17	0.60 25	*0.80* *34*	0.90 38
				γ	
Classifier	FP-rate FP lab tests	0.02 1 18	0.04 2 27	*0.10* *6* *40*	0.25 15 53
				γ	
Random selection	FP-rate FP lab tests	0.40 23 40	0.60 35 60	*0.80* *46* *80*	0.90 52 90
lab tests$_{random}$ / lab tests$_{classifier}$		2.22	2.22	*2.00*	1.70

Fig. 5. ROC-curve. A) Average ROC curve of the ten CV loops (ldc, backward feature selection). The light gray curves are the ROC curves of the separate CV loops. The diagonal line illustrates the random selection ROC curve. **B)** Numeric example that shows the amount of lab work that could be saved for different operating points.

Interestingly, when using the optimal parameter settings, the raw signal of the hydrophilic peaks is not smoothed. With *window size* = 3 and *edge* = 0.0, the value at a specific location in the sequence is simply the amino acid scale value of the amino acid at that specific location. Therefore, the feature is actually the same as the GRAVY feature, but using an amino acid scale in which all values greater than the threshold are set to zero, and all other values are set to the threshold minus the value. In this case, arginine is set to 0.1, lysine to 0.33, and the rest of the amino acids is set to zero. From another perspective, this feature can be seen as an amino acid set composition for the set {arginine, lysine} in which the arginine has a higher weight.

It is questionable if the resulting feature is still related to the proteins hydrophilic character. Since both arginine and lysine are also basic amino acids, it could just as well be related to the proteins basic character. Furthermore, because of the small window size, the feature does not take into account sequence order. However, it could be hypothesized that hydrophilic amino acids will mainly contribute to the proteins hydrophilic character when they have a relatively high occurrence within a larger region.

3.3 Feature Correlation

Fig. 7 shows a heat map of the hierarchical clustered (complete linkage) feature correlation matrix. The cluster at the top left shows relatively high correlations, which can be explained by the fact that the features contain redundant data: *arginine* is part of both *basic* and *charged*, *basic* is a subset of *charged*, the isoelectric point is derived from a proteins charge and therefore correlated with *charged*, and *hydrophilic peaks* takes into account the amino acids arginine and

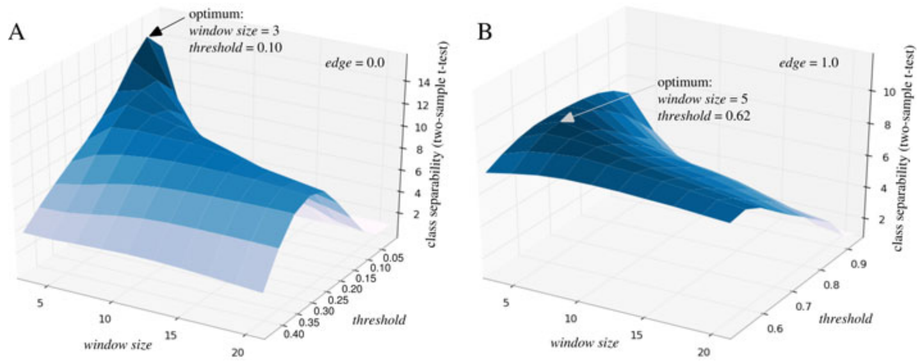


Fig. 6. Parameter optimization of hydropathic peaks features. A) Class separability scores for the hydrophilic peaks feature plotted against different parameter settings. **B)** The same as in *A*, but for the hydrophobic peaks feature. Both plots show the result for one *edge* value, different *edge* values provided similar plots. Both plots were obtained in one of the CV loops, the same optimum was found in all CV loops.

lysine, that are both in *basic* and *charged*. There is also a high correlation between *small*, *turn*, and *tiny*. This can also be explained by data redundancy: both *turn* and *tiny* are a subset of *small*.

3.4 Feature Selection

Using ldc with forward feature selection, the feature selection results of the 10 CV loops showed that: *asparagine* was always part of the top-3 selected features (7 times selected first), either *hydrophilic peaks* or *basic* was part of the top-3 selected features 9 times (6 times selected second), *hydrophobic peaks* was part of the top-4 selected features 9 times (7 times selected third), and *tyrosine* was part of the top-4 selected features 6 times (5 times selected fourth).

The high correlation between *hydrophilic peaks* and *basic* (Fig. 7), together with the fact that both have a high class separability score (Table 1), explains their mutual exclusive selection. In Fig. 4, the colors above the feature names depict what features are in the same correlation cluster and the arrows indicate what features are most often in the top-4 selected features. It shows that these features are in different correlation clusters, and are the best performing ones of their cluster. Therefore, feature selection seems to select individual features that best represent an underlying cluster of related features.

4 Discussion

To be useful for large-scale production, a protein should be produced and secreted with high yield. We report a sequence-based approach to classify proteins into *successful* or *unsuccessful* production, which was trained and validated on a set of 638 proteins. We used 10-fold CV for feature selection and classifier

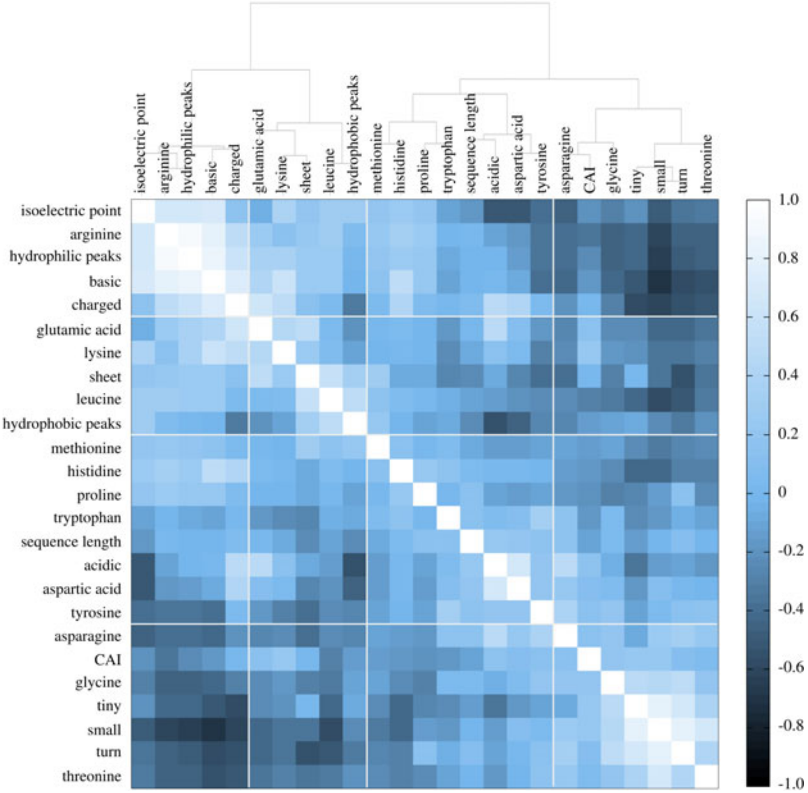


Fig. 7. Heat map of clustered feature correlation matrix

training to avoid biased performance results. Since we are mostly interested in the operating points of the first 30 percent of the ROC-curve, we used the AUC of this region as the main performance measure.

We calculated 39 features and used the 25 with highest class separability score for classification. We showed that both a classifier that uses all features and a classifier trained with feature selection, outperform classifiers trained on single features. The classifiers trained with feature selection did not significantly outperform the classifier trained on all 25 features, indicating that all features contribute to the result.

Furthermore, the feature selection results showed that asparagine, the set {arginine, lysine}, and tyrosine, as well as the hydrophobic peaks feature, were most defining in case of the linear discriminant classifier. To get more insight into protein secretion, it would be interesting to link the biological significance of these features to protein secretion mechanisms. For example, the asparagine composition could be related to N-linked glycosylation, a process that in many cases is important for protein folding and stability [10].

Acknowledgments. This work was part of the BioRange programme of the Netherlands Bioinformatics Centre (NBIC), which is supported by a BSIK grant through the Netherlands Genomics Initiative (NGI). The data set was provided by DSM Biotechnology Center.

References

1. Benita, Y., Wise, M., Lok, M., Humphery-Smith, I., Oosting, R.: Analysis of high throughput protein expression in Escherichia coli. Mol. Cell. Proteomics 5(9), 1567 (2006)
2. Chang, C., Lin, C.: LIBSVM: a library for support vector machines (2001)
3. Duin, R., Juszczak, P., Paclik, P., Pekalska, E., de Ridder, D., Tax, D., Verzakov, S.: A Matlab toolbox for pattern recognition. PRTools version 4.1, 3 (2000)
4. Horton, P., Park, K., Obayashi, T., Fujita, N., Harada, H., Adams-Collier, C., Nakai, K.: WoLF PSORT: protein localization predictor. Nucleic Acids Res. 35(Web Server issue), W585–W587 (2007)
5. Klee, E., Sosa, C.: Computational classification of classically secreted proteins. Drug Discovery Today 12(5-6), 234–240 (2007)
6. Kurgan, L., Razib, A., Aghakhani, S., Dick, S., Mizianty, M., Jahandideh, S.: CRYSTALP2: sequence-based protein crystallization propensity prediction. BMC Struct. Biol. 9, 50 (2009)
7. Kyte, J., Doolittle, R.: A simple method for displaying the hydropathic character of a protein. J. Mol. Biol. 157(1), 105–132 (1982)
8. Magnan, C., Randall, A., Baldi, P.: SOLpro: accurate sequence-based prediction of protein solubility. Bioinformatics 25(17), 2200–2207 (2009)
9. Matthews, B.: Comparison of the predicted and observed secondary structure of t4 phage lysozyme. BBA-Protein Struct. 405(2), 442–451 (1975)
10. Mitra, N., Sinha, S., Ramya, T., Surolia, A.: N-linked oligosaccharides as outfitters for glycoprotein folding, form and function. Trends Biochem. Sci. 31(3), 156–163 (2006)
11. Nevalainen, K., Te'o, V., Bergquist, P.: Heterologous protein expression in filamentous fungi. Trends Biotechnol. 23(9), 468–474 (2005)
12. Nielsen, H., Engelbrecht, J., Brunak, S., Von Heijne, G.: Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. Protein Eng., Des. Sel. 10(1), 1 (1997)
13. Pel, H., de Winde, J., Archer, D., Dyer, P., Hofmann, G., Schaap, P., Turner, G., de Vries, R., Albang, R., Albermann, K., et al.: Genome sequencing and analysis of the versatile cell factory Aspergillus niger CBS 513.88. Nat. Biotechnol. 25(2), 221–231 (2007)
14. Pierleoni, A., Martelli, P., Fariselli, P., Casadio, R.: BaCelLo: a balanced subcellular localization predictor. Bioinformatics 22(14), e408–e416 (2006)
15. Sharp, P.M., Li, W.H.: The codon adaptation index - a measure of directional synonymous codon usage bias, and its potential applications. Nucleic Acids Res. 15(3), 1281 (1987)
16. Tsang, A., Butler, G., Powlowski, J., Panisko, E., Baker, S.: Analytical and computational approaches to define the *Aspergillus niger* secretome. Fungal Genet. Biol. 46(1), S153 (2009)
17. Wessels, L., Reinders, M., Hart, A., Veenman, C., Dai, H., He, Y., van't Veer, L.: A protocol for building and evaluating predictors of disease state based on microarray data. Bioinformatics 21(19), 3755–3762 (2005)

Machine Learning Study of DNA Binding by Transcription Factors from the LacI Family

Gennady G. Fedonin and Mikhail S. Gelfand

Institute for Information Transmission Problems (the Kharkevich Institute), RAS
Bolshoy Karetny per. 19, Moscow, 127994, Russia
gennady.fedonin@gmail.com,
gelfand@iitp.ru
http://www.rtcb.iitp.ru

Abstract. We studied 1372 LacI-family transcription factors and their 4484 DNA binding sites using machine learning algorithms and feature selection techniques. The Naive Bayes classifier and Logistic Regression were used to predict binding sites given transcription factor sequences. Prediction accuracy was estimated using 10-fold cross-validation. Experiments showed that the best prediction of nucleotide densities at selected site positions is obtained using only a few key protein sequence positions. These positions are stably selected by the forward feature selection based on the mutual information of factor-site position pairs.

Keywords: transcription factors, naive Bayes classifier, logistic regression, mutual information.

1 Introduction

Many biological processes involve specific interaction between DNA-binding proteins and DNA sites. The mechanisms of the sequence- and structure-specific recognition remain elusive, despite some advance coming from experimental mutagenesis studies and computational analysis of known X-ray structures of protein-DNA complexes [1], [2]. One of the reasons for that may be lack of data. Indeed, while many complexes are structurally resolved, one of the main results of the analysis has been the absence of a universal protein-DNA recognition code [3], [4], [5]. On the other hand, experimental analysis has been limited to a small number of proteins, and again, the obtained results do not seem universal [6].

A different approach is to study the protein-DNA code within large families of DNA-binding proteins [7], e.g. C2H2 zinc finger, homeodomain and bHLH domains [8] or TAL receptors [9]. At that, the data may come not only from experiment, but from comparative genomic analysis of regulatory interactions. A rich source of such data are bacterial transcription factors, e.g. the LacI family considered here. Given the data on sites bound by given proteins, one may study correlations between the amino acid sequences and corresponding DNA sites, and then to use the structures, if known, as a sanity check, verifying that the observed positions indeed form contacts in the protein-DNA complexes.

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 15–26, 2010.

One observation coming from early studies [10] has been that the correlations are not limited to pairs of positions in the protein and DNA alignment: in many cases the protein preferences to a particular nucleotide at a particular site position seemed to depend on specific residues at several protein positions. This leads to the problem of selecting the optimal model complexity. Here we address this problem using the predictive power of pattern recognition algorithms as a tool to determine the optimal number of the model parameters.

2 Materials and Methods

2.1 Data

The LacI-family bacterial transcription factor and their binding sites were selected from the LACI_DB database (O. Laikova, unpublished). The DNA-binding domain (HTH_LACI) boundaries for each protein were determined using SMART_DB [11]. The obtained sequences were aligned against the standard HTH_LACI domain alignment with minimal manual editing, resulting in an alignment of 1372 protein sequences. The resulting alignment length was 87 positions. Sixteen positions with more than 30% gaps were removed. The sample of DNA sites contained 4484 sequences. The data may be downloaded from the RegPrecise database [12].

Hence, we had a sample of protein-site pairs, and the aim was to predict the probability density of nucleotides at site positions given the protein amino acid sequence (AAS). We assumed all site positions to be mutually independent given AAS, hence each position was predicted separately.

2.2 Cross-Validation

To estimate the prediction accuracy, the initial sample was randomly split into ten sets, each of which was used as a testing set with training on the remaining nine sets. Since many proteins in the sample are closely related (and have very similar AAS) it is reasonable to require the testing set not to contain AASs too similar to an AAS in the training set. To ensure this, we grouped similar AASs by similarity into clusters never separated during splitting. At that, we calculated pairwise similarity (percent of identical amino acid) for all AAS pairs. Next, we built a full graph with AASs as vertices and edges weighted with the similarity values, and removed all edges with weight less than a fixed threshold. The similarity clusters were defined as maximal connected components.

For each split into test and training sets, all algorithms were trained and their log-likelihoods on the testing set were calculated. Log-likelihood was calculated as:

$$logL = \frac{\sum_i w_i \sum_j P(n_{ij}|S_i)}{\sum_i w_i} \ ,$$

where index i runs over all AAS, index j runs over all sites of the i–th AAS, n_{ij} is the nucleotide observed at the selected position of the j–th site of the i–th sequence, w_i is the weight of the i–th AAS. The results were averaged. The procedure was repeated ten times for better averaging.

2.3 Algorithms

Weighting amino acid and binding site sequences. The similarity clusters vary in size with some sequence motifs being overpresented. To compensate for this, protein sequences were weighted, so that closely related proteins were assigned smaller weights than proteins different from all others, using the Gerstein-Sonhammer-Chotia algorithm. Each protein weight was divided equally among all its binding sites, resulting in weights of AAS-site pairs.

These weights were used to compute amino acid residue and nucleotide frequencies for building the Bayesian classifier, computation of the mutual information, and for training the logistic regression.

Naive Bayes classifier. The Bayesian classifier [13] estimates the occurrence probability for each nucleotide at each site position using the Bayes formula:

$$P(n_i|S) = \frac{P(n_i)P(S|n_i)}{\sum_j P(n_j)P(S|n_j)} \ ,$$

where n_i is the i-th nucleotide, S is the amino acid sequence, $P(n)$ is the prior probability of nucleotide n.

The naive Bayes approach assumes all positions in AAS to be mutually independent given site position nucleotide:

$$P(S|n) = \prod_i P(a_i|n) \ ,$$

where a_i is the amino acid residue at position i. Probabilities $P(a_i|n)$ are estimated using the corresponding frequencies in the sample, with phylogenetic weights and pseudocounts.

Logistic regression. The logistic regression [14] is a popular machine learning algorithm for two-class classification tasks. The training objects are assumed to be numerical feature vectors with $\{-1, 1\}$ labels. The algorithm builds a linear decision rule, weighting each numerical feature:

$$f(x_1, \ldots, x_n) = sign\Big(\sum_{i=1}^{n} \alpha_i x_i\Big) \ ,$$

or in the vector form:

$$f(\boldsymbol{x}) = sign(\langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle) \ ,$$

where α_i is the weight of i-th feature, x_i is the value of the i-th feature.

Learning is performed by searching for weights that optimize the quality function on the training set:

$$L(\boldsymbol{\alpha}) = \sum_{i=1}^{l} w_i \ln \sigma(y_i \langle \boldsymbol{\alpha}, \boldsymbol{x}_i \rangle) - k \sum_{i=1}^{n} \alpha_i^2 \to max_{\boldsymbol{\alpha}} \ ,$$

where index i runs over all training objects, $y_i \in \{-1, 1\}$ is the class of the i-th object, w_i is the weight of the i-th object , $\sigma(z) = \frac{1}{1+\exp(-z)}$ is the logistic (sigmoid) function, $k \sum_{i=1}^{n} \alpha_i^2$ is a regularization term, k is an a priori fixed regularization parameter.

Class probabilities given feature vector can be estimated using the sigmoid function:

$$P(y) = \frac{1}{1 + \exp(-y\langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle)} \ ,$$

where $y \in \{-1, 1\}$ is the class value, $\boldsymbol{x}$ is the feature vector, $\boldsymbol{\alpha}$ is the weight vector.

The logistic regression requires numeric features. In our case all features are nominal. We used the standard binarization approach: each amino acid residue a_k at i-th position was mapped to an indicator binary feature:

$$f_i(a) = \begin{cases} 1, & \text{when } a = a_k \ ; \\ 0, & \text{otherwise } . \end{cases}$$

To predict four nucleotide probabilities, an individual classifier was trained for each nucleotide. ASS-site pairs with a given nucleotide at the given site position were used as positive training examples, all other pairs, as negative ones. The positional probability of each nucleotide was calculated as:

$$P(n_i|S) = \frac{P_i(+|S)}{\sum_{j=1}^{4} P_j(+|S)} \ ,$$

where S is the AAS for which predictions are made, $P_i(+|S)$ is the positive class probability computed by i-th classifier.

The weights for the negative objects were set to the weight of the corresponding AAS, and for positive objects, the same weight, multiplied by the frequency of the given nucleotide at the given site position.

Feature selection using mutual information. The mutual information (MI, [15]) of the AAS-site position pair is the measure of correlation of these positions, allowing for a quick estimation of the predicting power of the AAS position for the nucleotide at the site position. Calculating the MI is fast, making it convenient for the feature selection.

To offset for unreliable estimations of the frequencies of rare residues and nucleotides (at a given position), we used pseudocounts, adding small values for rare events.

The effective frequency of residue a at position i was defined as:

$$f_i(a) = \frac{N_i(a) + k\frac{\sum_b N_i(b)P(b \to a)}{\sqrt{N}}}{N + k\sqrt{N}} \ ,$$

where $N_i(a)$ is the total weight of AASs with a in position i, N is the total weight of all AASs in the sample. The transition probabilities $P(b \to a)$ were obtained from BLOSUM60 [16].

The effective frequency of nucleotide n at position j was:

$$f_j(n) = \frac{N_j(n) + k\frac{\sum_m N_j(m)P(m\to n)}{\sqrt{N}}}{N + k\sqrt{N}} = \frac{N_j(n) + 0.25k\frac{\sum_m N_j(m)}{\sqrt{N}}}{N + k\sqrt{N}} \ ,$$

where $N_j(n)$ is the total weight of sites with n at position j, N is the total weight of the sample sites.

The observed effective frequency of 'amino acid - nucleotide' pair:

$$f^{\mathrm{o}}_{ij}(a,n) = \frac{N_{ij}(a,n) + k\sqrt{N}f^{\mathrm{e}}_{ij}(a,n)}{N + k\sqrt{N}} \ ,$$

where $N_{ij}(a,n)$ is the total weight of pairs with a at position i of the AAS and n at the site position j, N is the total weight of sample pairs, $f^{\mathrm{e}}_{ij}(a,n)$ is the expected effective frequency of pair (a,n) defined as

$$f^{\mathrm{e}}_{ij}(a,n) = f_i(a)f_j(n) \ ,$$

where $f_i(a)$ and $f_j(n)$ are the effective frequencies of residue a at position i and nucleotide n at position j, respectively.

The mutual information was computed as

$$I_{ij} = \sum_a \sum_n f^{\mathrm{o}}_{ij}(a,n) \log \frac{f^{\mathrm{o}}_{ij}(a,n)}{f^{\mathrm{e}}_{ij}(a,n)} \ .$$

Greedy forward feature selection. Another strategy for feature selection is searching through subsets of features, training algorithms using feature subsets on parts of the training set, estimating error on remaining objects and selecting the subset with the minimal error.

In practice, the exhaustive search is computationally intractable, so we used the greedy algorithm, successively adding each of the remaining features to the current best subset and selecting the feature which provides the best classifier. This feature then is added to the best-feature subset and the process is repeated.

The greedy strategy takes into account feature dependency, but still can lead to suboptimal subsets. On the other hand, this strategy is the fastest after the MI-based feature selection.

3 Results and Discussion

We report only the performance of two simple algorithms: Naive Bayes classifier (NB) [13] based on amino acid frequencies estimation and logistic regression (LR) [14] with simple AAS encoding to feature vectors. We also tried using amino acid pairs' frequencies (and corresponding binarisation) with these algorithms, but the prediction quality was the same. The reason of this might be the data sparseness, which makes it impossible to estimate frequencies of complex events robustly. We also tried linear SVMs with these feature vectors, but the performance was poor. SVM with a linear kernel is the fastest in training SVM algorithm, but it is very slow compared with NB and LR. Using SVMs based on nonlinear kernels for feature selection required computational resources not available for this study.

3.1 Selecting Site Alignment Positions

Different site positions can be predicted with different accuracy. In this study we used those site positions, for which significantly correlated AAS positions were found [10]. We used the mutual information to measure correlation. As one can see in the heat map in Fig. 1, significant correlations are observed for positions 5, 6, 7, 9 and the symmetric ones. Below we consider only these four positions.

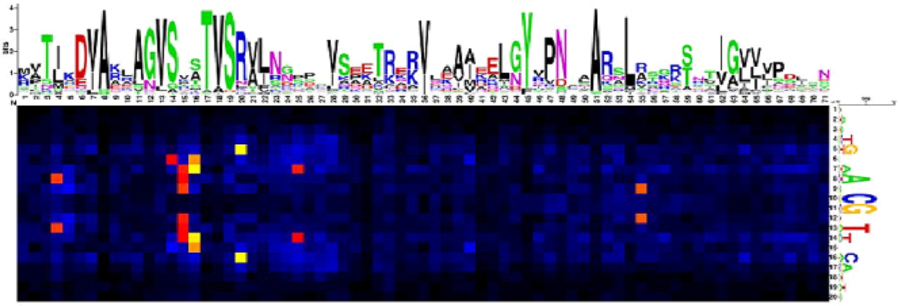


Fig. 1. Mutual information of AAS-site position pairs [10]. Light colors correspond to significant correlations.

3.2 Selection of Significant Positions

Selection was performed using two methods. Using the MI-based selection, twenty positions were selected for each of three site positions. Positions were selected successively, starting from the most informative one. On each iteration, the classifiers were trained using the current position set and the prediction quality (testing set log-likelihood) was estimated. The greedy selection was organized in the same way, but only for ten AAS positions for each site position. In both cases the process was repeated for different sample splits during 10-fold cross validation (2.2).

The prediction quality values for different feature set lengths were plotted on a graph. The selected positions were tabulated. The selected positions may vary for different sample splits. Hence we can only report the frequencies of given positions in position sets selected at algorithm iterations, i.e. the frequencies in the selected sets of sizes ranging from 1 to 20. To visualize the tables, we ordered all positions by the total frequency (the sum of frequencies in sets of all lengths) and report the top ones.

Only few positions are stably selected by both algorithms, i.e. these positions are selected with almost any sample split. The maxima of the test set log-likelihood plots often correspond to these position sets. Further increase of the position set size leads to overfitting. The selection stability and existence of well-defined maxima on the log-likelihood plots can be treated as a proof of connection between the selected AAS positions and the site positions.

While the prediction quality shows large variation, dependent on the split of the data into training and test sets, the overall results from different runs (position of the local maxima, selected positions, relative quality of predictions by different algorithms) are consistent.

3.3 AAS-Position Selection for Position 9 of the Site Alignment

The log-likelihood values obtained on the testing set for position 9 by various algorithms and selection strategies are plotted in Fig. 2. Well-defined maxima are obtained on three positions by all methods.

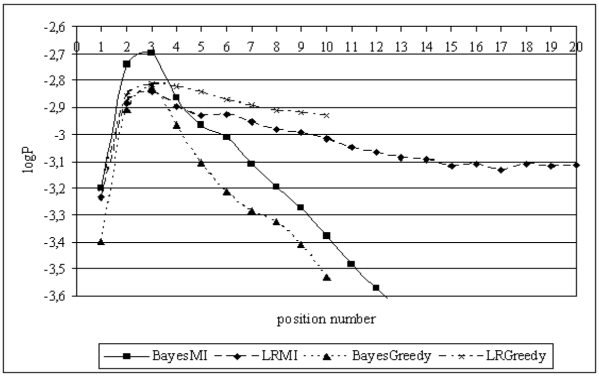


Fig. 2. The log-likelihood values against the number of selected positions for position 9 of the site alignment

Table 1 features the most frequent positions. The column numbers are the position numbers starting from the most frequent one. The row numbers are the selected set sizes. The MI-based search and greedy naive Bayes search stably select three positions 55, 15 and 5. The greedy logistic regression stably selects the same three positions, and frequently position 27.

The maximum prediction quality is achieved by using three positions. Therefore, positions 55, 15 and 5 of the amino acid alignment are significantly linked to position 9 of the site alignment.

3.4 AAS-Position Selection for Position 7 of the Site Alignment

The log-likelihood values obtained on the testing set for position 7 by various algorithms and selection strategies are plotted in Fig. 3. Well-defined maxima are

Table 1. Frequencies of six most frequent positions in MI-selected, greedy naive Bayes classifier (NB) and greedy logistic regression (LR) sets of varying lengths for prediction of site position 9 (in %)

	MI-selected						NB						LR					
Set size	55	15	5	68	56	16	55	15	5	1	70	26	55	15	5	27	49	56
1	100	0	0	0	0	0	91	9	0	0	0	0	96	4	0	0	0	0
2	100	100	0	0	0	0	100	100	0	0	0	0	100	100	0	0	0	0
3	100	100	90	0	0	0	100	100	96	0	0	0	100	100	90	9	0	0
4	100	100	90	20	35	39	100	100	99	36	5	6	100	100	96	82	5	4
5	100	100	95	50	64	57	100	100	99	52	23	23	100	100	98	94	38	37
6	100	100	97	79	80	80	100	100	99	68	42	40	100	100	99	96	64	54

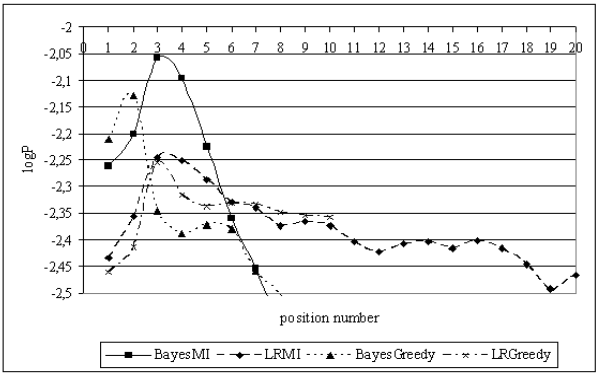


Fig. 3. The log-likelihood values against the number of selected positions for position 7 of the site alignment

obtained on three positions by all methods, except the greedy Bayes classifier, which has maximum on two positions.

The most frequent positions are listed in Tab. 2, with the notation as above. The MI-based search stably selects three positions 16, 25 and 15, and sometimes position 68. The greedy logistic regression stably selects the same three positions, whereas the greedy Bayes classifier based search makes a mistake on the third step, stably selecting position 49, which, as seen on the log-likelihood plot, leads to a dramatic decrease of the prediction quality.

The maximum prediction quality is achieved by using three positions. Therefore, positions 16, 25 and 15 of the amino acid alignment are significantly linked to position 7 of the site alignment.

3.5 AAS-Position Selection for Position 6 of the Site Alignment

The log-likelihood values are plotted in Fig. 4. The naive Bayes classifier with the MI-based selection has two maxima at one and three positions, while the greedy strategy has maxima at one and seven positions. The logistic regression

Table 2. Frequencies of six most frequent positions in MI-selected, greedy naive Bayes classifier (NB) and greedy logistic regression (LR) sets of varying lengths for prediction of site position 7 (in %)

	MI-selected						NB						LR					
Set size	16	25	15	68	5	46	16	15	49	68	50	19	16	15	25	49	68	50
1	100	0	0	0	0	0	100	0	0	0	0	0	100	0	0	0	0	0
2	100	96	4	0	0	0	100	97	0	2	0	0	100	69	30	0	3	0
3	100	100	100	0	0	0	100	97	71	11	9	2	100	99	99	0	0	0
4	100	100	100	84	5	3	100	98	89	59	33	7	100	100	100	56	20	5
5	100	100	100	94	25	18	100	98	92	94	75	12	100	100	100	78	57	16
6	100	100	100	97	38	46	100	99	93	100	86	64	100	100	100	89	84	42

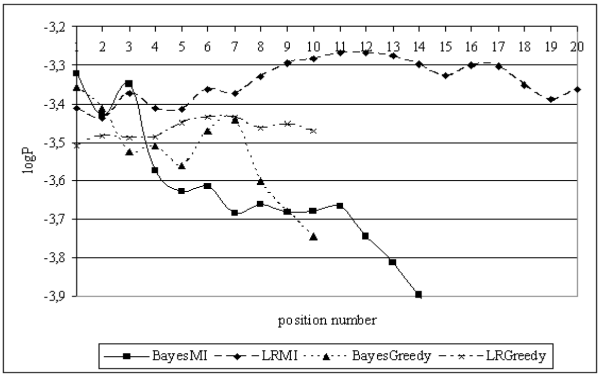


Fig. 4. The log-likelihood values against the number of selected positions for position 6 of the site alignment

curves slowly grow, having many local maxima with highest values around six and eleven positions for the greedy and MI-based search, respectively.

Table 3 features the most frequent positions. The MI-based selection has one absolutely stable position, 16, and two additional stable positions, 25 and 15, which are interchangeable at the second selection step. The greedy strategies select two positions, absolutely stable 16 and strongly stable 15. Further selection is unstable.

In prediction of position 6 in binding sites, different algorithms behave differently: the naive Bayes classifier has two maxima, while the logistic regression seems to overfit. However, all methods stably select position 16 of the AAS alignment that is significantly connected with position 6 in the site alignment.

3.6 AAS-Position Selection for Position 5 of the Site Alignment

The log-likelihood values obtained on the testing set for position 5 by different algorithms and selection strategies are plotted in Fig. 5. For the naive Bayes classifier, both MI-based and greedy, the maximum is reached when only one

Table 3. Frequencies of five most frequent positions in MI-selected, greedy naive Bayes classifier and greedy logistic regression sets of varying lengths for prediction of site position 6 (in %)

	MI-selected					NB					LR				
Set size	16	25	15	68	26	16	15	20	27	49	16	15	27	25	49
1	100	0	0	0	0	100	0	0	0	0	100	0	0	0	0
2	100	60	40	0	0	100	90	0	10	0	100	85	5	8	0
3	100	96	91	0	8	100	94	61	28	6	100	93	65	19	4
4	100	98	95	45	29	100	94	82	64	21	100	95	78	35	22
5	100	100	97	66	58	100	97	89	82	68	100	98	86	55	47

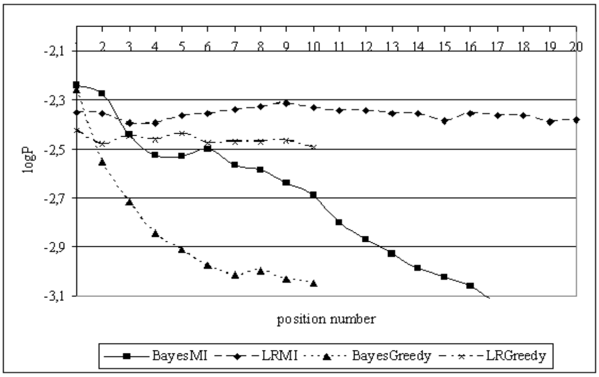


Fig. 5. The log-likelihood values against the number of selected positions for position 5 of the site alignment

Table 4. Frequencies of five most frequent positions in MI-selected, greedy naive Bayes classifier and greedy logistic regression sets of varying lengths for prediction of site position 5 (in %)

	MI-selected					NB					LR				
Set size	20	25	27	68	16	20	27	15	69	50	20	25	16	50	27
1	100	0	0	0	0	100	0	0	0	0	100	0	0	0	0
2	100	95	3	2	0	100	55	2	20	0	100	54	33	0	13
3	100	96	35	41	21	100	61	28	58	18	100	87	69	16	25
4	100	99	62	62	53	100	62	48	60	28	100	94	73	60	44
5	100	99	85	83	77	100	64	67	61	49	100	100	75	85	62

position is used for prediction. The logistic regression algorithm plots do not have a marked maximum.

The most frequent positions are tabulated in Tab. 4. Position 20 is absolutely stable, position 25 is stable for the MI-based search. Further selection is unstable.

The maximum prediction quality is achieved by using only one position and addition of the second position considerably decreases it. Therefore, only position 20 of the amino acid alignment is significantly connected with position 5 of the site alignment.

4 Conclusions

Experiments showed that knowledge of only a few key protein sequence positions is sufficient for prediction of nucleotide densities at selected site positions. These positions form significantly correlated pairs with corresponding site alignment positions, having high mutual information values. Moreover, the selected pairs of positions are largely the same for different methods (for any given site position) and correspond to the contacts in protein-DNA complexes [10]. On the other hand, the results show that the dependencies are not limited to simple pairs of contacting positions. Overall, these observations support the existence of protein family-specific protein-DNA recognition code. Analysis of other transcription factor families will show what features of this code are universal.

Acknowledgements

We are grateful to Olga Laikova and Dr. Alexandra B. Rakhmaninova who provided the data, and to Yuri Korostelev for sharing preliminary results and providing Fig. 1. This study was partially supported by RAS (program "Molecular and Cellular Biology"), RFBR (grains 09-04-92745 and 10-04-00431), and state contract 2.740.11.0101.

References

1. Luscombe, N.M., Laskowski, R.A., Thornton, J.M.: Amino acid-base interactions: a three-dimensional analysis of protein-DNA interactions at an atomic level. Nucleic Acids Research 29, 2860–2874 (2001)
2. Baker, C.M., Grant, G.H.: Role of aromatic amino acids in protein-nucleic acid recognition. Biopolymers 85, 456–470 (2007)
3. Suzuki, M., Brenner, S.E., Gerstein, M., Yagi, N.: DNA recognition code of transcription factors. Protein Engineering 8, 319–328 (1995)
4. Benos, P.V., Lapedes, A.S., Stormo, G.D.: Is there a code for protein-DNA recognition? Probab(ilistical)ly. Bioessays 24, 466–475 (2002)
5. Luscombe, N.M., Thornton, J.M.: Protein-DNA interactions: amino acid conservation and the effects of mutations on binding specificity. Journal of Molecular Biology 320, 991–1009 (2002)
6. Luscombe, N.M., Austin, S.E., Berman, H.M., Thornton, J.M.: An overview of the structures of protein-DNA complexes. Genome Biology 1, REVIEWS001 (2000)
7. Sandelin, A., Wasserman, W.W.: Constrained binding site diversity within families of transcription factors enhances pattern discovery bioinformatics. Journal of Molecular Biology 338, 207–215 (2004)

8. Mahony, S., Auron, P.E., Benos, P.V.: Inferring protein-DNA dependencies using motif alignments and mutual information. Bioinformatics 23, i297–i304 (2007)
9. Moscou, M.J., Bogdanove, A.J.: A simple cipher governs DNA recognition by TAL receptors. Science 326, 1501
10. Korostelev, Y., Laikova, O.N., Rakhmaninova, A.B., Gelfand, M.S.: Correlations between amino acid sequences of transcription factors and their DNA binding sites. In: Abstr. First RECOMB Satellite Conference on Bioinformatics Education, San Diego, USA (2009)
11. Kalinina, O.V., Novichkov, P.S., Mironov, A.A., Gelfand, M.S., Rakhmaninova, A.B.: SDPpred: a tool for prediction of amino acid residues that determine differences in functional specificity of homologous proteins. Nucleic Acids Research 32, W424–W428 (2004)
12. Novichkov, P.S., Laikova, O.N., Novichkova, E.S., Gelfand, M.S., Arkin, A.P., Dubchak, I., Rodionov, D.A.: Nucleic Acids Research 38, D111–D118 (2010)
13. Domingos, P., Pazzani, M.: On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29, 103–137 (1997)
14. Hosmer, D., Lemeshow, S.: Applied Logistic Regression, 2nd edn. Wiley, Chichester (2000)
15. Peng, H.C., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 1226–1238 (2005)
16. Henikoff, S., Henikoff, J.G.: Amino Acid Substitution Matrices from Protein Blocks. Proc. Natl. Acad. Sci. USA 89, 10915–10919 (1992)

Joint Loop End Modeling Improves Covariance Model Based Non-coding RNA Gene Search

Jennifer Smith

Electrical and Computer Engineering Department, Boise State University,
1910 University Drive, Boise, Idaho 83725-2075, USA
JASmith@BoiseState.edu

Abstract. The effect of more detailed modeling of the interface between stem and loop in non-coding RNA hairpin structures on efficacy of covariance-model-based non-coding RNA gene search is examined. Currently, the prior probabilities of the two stem nucleotides and two loop-end nucleotides at the interface are treated the same as any other stem and loop nucleotides respectively. Laboratory thermodynamic studies show that hairpin stability is dependent on the identities of these four nucleotides, but this is not taken into account in current covariance models. It is shown that separate estimation of emission priors for these nucleotides and joint treatment of substitution probabilities for the two loop-end nucleotides leads to improved non-coding RNA gene search.

Keywords: Sequence analysis, RNA gene search, covariance models.

1 Introduction

Covariance models are an effective method of capturing the joint probability information inherent in the intramolecularly base-paired positions of a non-coding RNA molecule [1, 2]. Unlike profile hidden Markov models [3, 4], which have a set of four emission probabilities over the possible nucleotides at each consensus sequence position, covariance models allow consensus base pairs to be assigned sixteen joint probabilities over the possible ordered nucleotide pairs. Covariance models also allow the probability of insertion or deletion of a base pair to be different than the sum of the marginal probabilities of insertion or deletion of the individual nucleotides. The profile hidden Markov model can be viewed as a special form of a covariance model with no base pairs specified.

Covariance models are finite state machines which require the estimation of state emission and state transition probabilities as well as model structure. This is normally done using a family of known sequences in a multiple alignment with secondary structure annotation. Counts of nucleotide frequencies in unpaired consensus columns or nucleotide pair frequencies in couples of base-paired consensus columns form the basis for emission probabilities. Counts of missing nucleotides in consensus columns and of nucleotide presence in non-consensus columns can be used to generate transition probabilities in and out of deletion and insertion states respectively.

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 27–37, 2010.

Conceptually, estimation of emission and transition probabilities is as simple as calculating the observed frequency of occurrence in the multiple alignment. The reality is much more complex. The very small number of family sequences that most RNA family models are estimated from is a major problem. In the *Rfam* 9.1 (December 2008) database of RNA alignments and covariance models, more than half of the 1371 family models are estimated from ten or fewer sequences [5, 6]. Most of the possible mutations, insertions, or deletions are never observed even though we have no particular reason to believe that they should be excluded from consideration. At very least pseudocounts need to be added to all possibilities such that the probability estimates do not outright exclude them. Pseudocounts are a form of prior information used in the estimation.

Far more informative priors than simple pseudocounts are needed for effective estimation of family models formed from so few sequences. Generic mutation, insertion, and deletion probabilities are obtained via observed frequency from the entire database of all RNA families. The generic emission and transition probabilities are found separately for base-paired and non-base-paired positions and with dependence on whether adjacent positions are paired or not. It will be demonstrated that these classifications are not quite fine enough later in this paper. In order to automatically uncover groups of mutation, deletion, or insertion patterns that tend to be observed together, these generic priors are estimated as a Dirichlet mixture [7] in recent versions of the Infernal [8] suite of programs for covariance-model-based RNA family analysis and search.

When combining the observed-frequency information from the multiple alignment of a specific family with the generic prior information, it is necessary to obtain a weighting based on our confidence in the family specific data versus our generic information. Having more sequences in the specific family increases our confidence in that data. However, simple counts of number of sequences are not very effective because our set of known sequences is rarely a random sample of actual sequences from the true complete family. We may have many sequences that are nearly identical and only a few with lots more diversity. This causes a simple count of number of sequences to overestimate the true information content. The usual solution to this problem is to employ entropy weighting based on the variability of the known family sequences [9].

There is a large literature on RNA secondary structure estimation based on primary sequence [10, 11]. Much of this literature uses the results of laboratory thermodynamic studies of RNA as its basis. These thermodynamic measurements are not used in covariance-model-based RNA family modeling. Instead, observed mutations, insertions, and deletions within the family or over the entire database (the priors) are used. However, it may be useful to study the regularities in RNA free energy measurements in the laboratory to guide choices in how covariance models are constructed. From the laboratory, we know that the identities of the nucleotides at the interface between the stem and the loop of a hairpin structure greatly affect thermodynamic stability of the hairpin structure. We also know that the length of the loop is a factor in stability. The mechanisms to capture these regularities are weak and nonexistent, respectively, in current covariance modeling practice. This paper will examine the stem/loop interface, but not loop length.

Some initial evidence that interface nucleotides and loop length might be important was found by Smith and Wiese [12]. This paper presents much more evidence for the stem/loop interface. It also looks at implementing a new type of node in the covariance model that can get around some of the problems encountered in tricking the existing Infernal program suite into handling the loop end nucleotides jointly.

The next section will review covariance models and estimation of model parameters in more detail. Section 3 looks at the regularities in free energy change when forming RNA hairpins observed in the laboratory. Changes to covariance model structure and parameter estimation procedure that can capture the observed thermodynamic regularities is presented in Section 4. Results of computational experiments on data from the Rfam database are presented in Section 5, followed by conclusions.

2 Covariance Model Structure and Parameter Estimation

Covariance models are finite state machines composed of emitting and silent states and directed edges connecting some of the states to some of the others. There is a unique starting state (called the root start state) and one or more terminal states (called end states). Given any nucleotide sequence it is possible to find the most probable mapping of the sequence onto model state visits and the associated overall probability of this mapping. Given a family of sequences, it is possible to find a set of state emission and state transition probabilities such that the overall probability when mapping a family member to the model is high and of mapping a dissimilar sequence to the model is low.

2.1 Model Structure

The states of a covariance model and the connectivity of these states can be determined from a consensus secondary structure of the RNA family. RNA secondary structure is a listing of pairs of sequence positions that intramolecularly base pair. The state structure can be described at a high level through the use of node trees, where nodes of a given class have identical internal state structure.

Figure 1 shows an example of a consensus secondary structure for an RNA family (right). The letters refer to the consensus nucleotides and the subscripts to the consensus sequence positions. The figure also shows the covariance model node tree for the same secondary structure. S, B, and E-type nodes contain no consensus emitting states. L and R-type nodes contain a single-emission consensus state and P-type nodes contain a pair-emission consensus state. The model is entered at the root start state located in the S0 node and has two exit points at the end states contained in nodes E12 and E22.

The node tree is simply a guide for constructing the underlying state model. The state model is the final model of interest. Figure 2 shows internal state structure of some of the nodes from the node tree in Figure 1. Nodes of the same type have the same internal structure, so constructing the state machine from the node tree is straightforward. There is a standard rule for how to connect edges from states in one

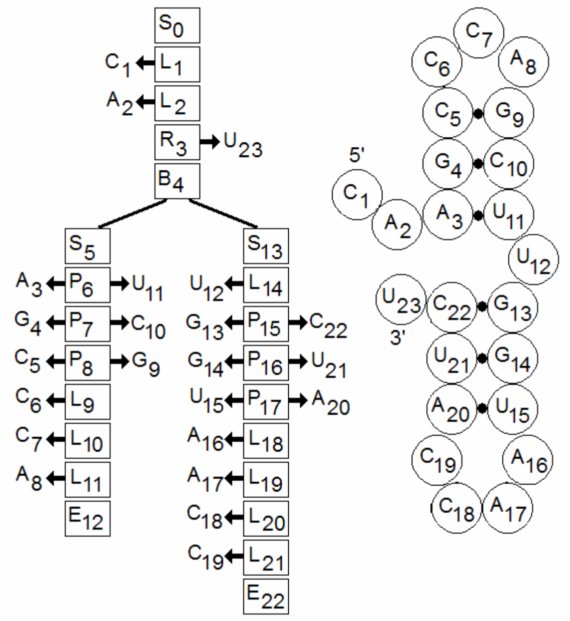


Fig. 1. An example consensus RNA secondary structure (right) and associated covariance model node tree (left)

node to states in an adjacent node. Each node contains one consensus state and varying numbers of non-consensus states. P, L, R, IL, and IR states types are emitting and all others are silent. D states allow for deletions relative to the consensus and IL or IR states allow for insertions.

2.2 Model Parameters

Once we have state structure, it is necessary to estimate emission probabilities for emitting states and transition probabilities for each edge connecting states. These probabilities are converted to log-likelihood ratios so that the total (log) probability of a particular path can be computed as the sum of transition and emission probabilities along the path. Dynamic programming can then be used to find the most probable path for a given sequence.

The parameters are estimated through a weighted combination of observed frequency of events in the family multiple alignment and the prior for the parameter. The priors in turn depend on the type of node holding the state and on adjacent node types. As an example, transition probabilities into and out of the D state in the R3 node at the top of Figure 2 would depend in part on the count of the number of gap characters in the twenty-third consensus column of the family multiple alignment. The R state in the R3 node is the consensus state which emits a consensus U and the D state in the R3 node is used to bypass this emission when a sequence has a deletion at this position relative to the consensus. Even though U is the consensus nucleotide for position 23, there are actually four emission probabilities associated with the R state in node R3. The probability for U is simply the highest of the four.

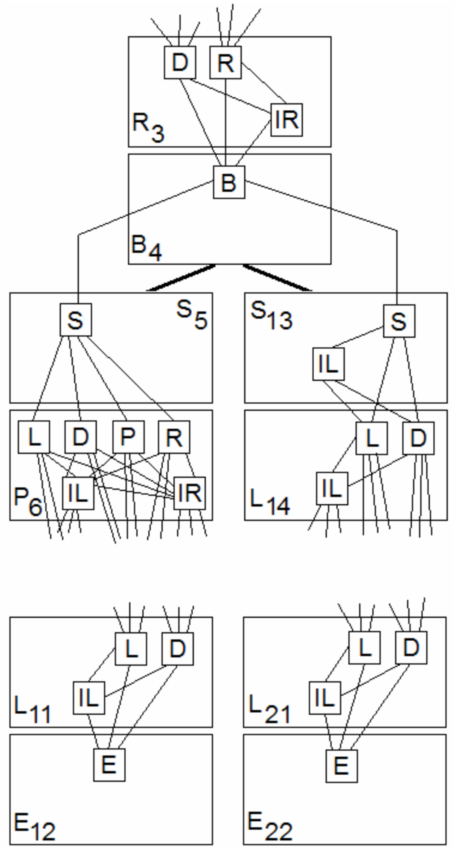


Fig. 2. Internal state structure of portions of the example covariance node tree from Figure 1

3 Thermodynamic Regularities

The thermodynamic stability of RNA hairpins is a fairly well studied topic [13-18]. Using calorimetry observations of the folding of short synthetic strands of RNA, models of the free energy of larger hairpin structures can be inferred. These models are used extensively in algorithms to predict secondary structure of RNA from sequence. These algorithms are based on the idea that the final conformation of an RNA molecule will be close to that of the minimum free-energy conformation.

Two of the major observations from the laboratory data is that hairpin stability depends on the number of nucleotides in the loop and on the identities of the four nucleotides at the stem-loop interface. The loop-length observation is relevant to covariance models and should be addressed, but the focus in this paper is on the stem-loop interface observation.

In Figure 3, the stem-loop interface is composed of the closing pair U15 and A20 as well as the loop ends A16 and C19. Although the structure appears symmetric in the figure, the free energy of the structure shown for GGUAACCAUC is different than its mirror CUACCAAUGG. In other words, it maters which side of the stem-loop interface is 5' and which is 3'. Covariance model P nodes can emit any of the sixteen possible ordered pairs of nucleotides. In the middle of a stem it makes sense to allow all sixteen possibilities since a mutation from a Watson-Crick or wobble base pair (a canonical base pair) to a non-canonical pair can be held together by adjacent base pairs in the stem without necessarily destroying the stem. If the closing pair becomes non-canonical, then effectively the loop length increases by two and the next pair up the stem becomes the closing pair. So, there are really only six consensus ordered pairs to consider for the closing pair: AU, UA, CG, and GC (Watson-Crick) as well as the wobble pairs GU and UG. In the Rfam database, consensus wobble pairs are very infrequent at the closing pair position (observed only about 4.1% of the time in version 8.1).

In the work of Vecenie and Serra [13] a number of regularities are noted regarding the thermodynamic stability of hairpin structures when different nucleotides are present in the stem-loop interface. They note that if the closing pair is CG or GC and loop ends are GA or UU (but not AG), then the hairpin is much more stable. They also note that if the closing pair has a purine (A or G) on the 5' side, the GG loop ends are particularly stable.

It is hypothesized here that some RNA families may not be able to function as well with less stability in one or more of their hairpins. If this is so, then it would be desirable to penalize database search scores when the database sequence implies a mutation away from one of the very stable consensus configurations noted above. Unfortunately, covariance model structure and parameter priors do not allow for these thermodynamic regularities to be expressed either directly or indirectly.

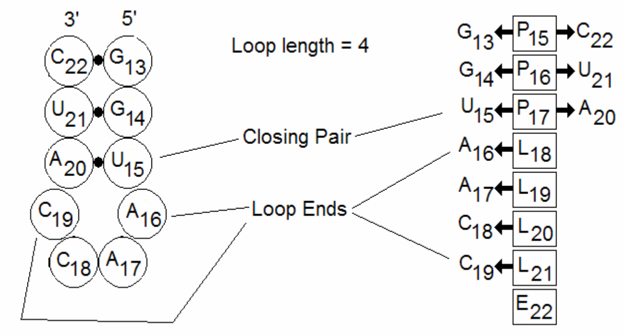


Fig. 3. A portion of the RNA secondary structure and covariance node tree from Figure 1 showing a single hairpin with the locations of the stem's closing pair and the loop ends labeled

4 Changes to Model Structure and Estimation

A major problem making expression of the thermodynamic regularities described in the previous section not possible is that the four nucleotides in the stem-loop interface are contained in three covariance model nodes with independent emission probabilities. Another problem is that the priors used for these emission probabilities are estimated as a mixture of database locations corresponding to stem-loop interfaces and to other structures.

To allow for expression of a regularity such as stable GG loop ends when the 5' side of a closing pair is A or G requires a new type of covariance model node. Such a node replaces a P node and two L nodes of a hairpin structure. In Figure 3, these are the P17, L18, and L21 nodes. Two hundred fifty six joint emission probabilities are needed for the consensus state of this node type. Since 160 of these combinations are not seen in practice (the combinations with non-canonical closing pairs), they can simply be assigned a very low probability, leaving only 104 emission probabilities that need to be estimated. Since wobble pairs are relatively rare, it may also be desirable to treat them as a class with a single emission probability (but a different value than for non-canonical pairs). This would leave 64 emission probabilities to be estimated for the Watson-Crick closing pairs. Clearly, heavy reliance on priors for these probabilities is needed since so few families have known sequences numbering in the hundreds and even fewer have enough variation in the observed stem-loop interface nucleotide combinations.

Implementation of a new node type requires significant programming effort to rewrite program suites such as Infernal. A partial solution is to at least express the joint probability of the two loop end nucleotides by tricking the existing algorithms. If the two loop-end L nodes are replaced by a single P node modeling these loop ends, expression of the joint probabilities of emission is possible. In Figure 3, the L18 and L21 nodes would be removed and replaced by a single P18 node directly below the existing P17 closing-pair node. In practice this can be accomplished simply by marking the two loop ends as if they were consensus base pairs in the input multiple alignment file to the *cmbuild* program of the Infernal program suite.

Using the P-node substitution trick does cause a couple of problems with priors. Firstly, The closing-pair P node will now use priors associated with a P node with P node child rather than the correct P node with L node child priors. This first problem can be solved by running the *cmbuild* program twice, once with and once without the loop ends marked as base paired. Then parameter estimates for the closing-pair P node in the second run are used in place of the estimates in the first run. The second problem is that the priors for the fake loop-end P node are completely wrong. The standard P node priors are generated from stem locations in the overall Rfam database with high probabilities for Watson-Crick base pairs, somewhat lower probabilities for wobble pairs and very low probabilities for non-canonical pairs. Instead, sets of priors for these loop-end P nodes are estimated on the side, one set for each possible consensus closing pair.

The loop-end P-node trick allows for a one-way dependence of loop-end emission probabilities on consensus closing pairs. It would be possible to also regenerate sixteen sets of priors for closing-pair P nodes and use the one associated with a given family's consensus loop ends. This two-way dependence would still not be quite as good as full use of joint probabilities.

5 Experimental Results

This section looks at results of using a P-node to model loop ends with non-standard priors on the loop-end P node only (and not for the closing pair P node).

First, the entire Rfam 8.1 database was processed and all 26,644 hairpin structures in all the seed sequences extracted. Since some RNA families have no hairpins and others have multiple hairpins, this number is different than the total number of seed sequences in the database. Table 1 shows the raw counts of number of observed loop-end pairs for each observed closing pair. Since wobble closing pairs are infrequent, they were not compiled separately, but are including the "All" column (such that the AU, UA, CG and GC columns do not add up to the All column). These raw counts are not that useful because the background frequencies of A, C, G and U are not each one quarter. To remedy this, Table 2 shows the same data as base-2 log-likelihood ratios. The log form is what is used by Infernal in order that the algorithm calculate additions instead of multiplications and it is visually useful since positive values are more likely than chance and negative less likely.

Some of the regularities noted in section 3 are apparent in Table 2. GA and UU loop ends are overrepresented by a factor of four when the closing pair is GC and by a factor of two when the closing pair is CG (but not for AU or UA closing pairs). Some other combinations have deviations of up to a factor of eight (for example UG loop ends on a UA closing pair).

The log-likelihood ratios of Table 2 were used as priors for loop-end P nodes on the fourteen shortest RNA families in the Rfam database which contained a hairpin without a pseudoknot. Pseudoknots are a situation where at least one pair of base pairs is such that neither base pair is completely between the other in sequence [19]. Covariance models use stochastic context-free grammars [20], which are incapable of describing a pseudoknot. Covariance models handle pseudoknots by treating some of the actually base-paired positions as if they were unpaired. Since what appears to be a hairpin in the node tree of pseudoknotted RNA families is actually something somewhat more complex, they will not be considered. The amount of computation time require to calculate E-values for covariance models is extremely high and goes up by more than the square of sequence length and short sequences are the most difficult to find in database search, so short sequences were chosen for this experiment.

Table 3 shows the results of the computational experiment. The first two columns show the length of the consensus sequence and the number of known family sequences. Both the seed sequences used to construct the family models and those found through database search by the curators of Rfam are included in this number. E-values are calculated by the Infernal program suite by reshuffling the known sequence many times (5000 times chosen for this study), scoring each reshuffled sequence against the family covariance model and then and fitting the resulting scores to a Gumble extreme value distribution [21]. The score of the unshuffled sequence is then used to find the probability of matching or exceeding the unshuffled score by pure chance. Lower E-values imply better specificity given that the threshold is set such that the sequence is just barely accepted as a true positive. The E-value ratios shown are the ratio of the E-value using the standard covariance model divided by the E-value with the loop-end P node. Ratios greater than one mean that using the loop-end P node has more power than the standard model. A E-value ratio of two means that we expected twice as many false alarms from the standard model.

On average, in only two cases (Rfam accession numbers RF00469 and RF00496) did modeling the loop ends jointly do significantly worse and in most cases it did quite a bit better.

Table 1. Counts of loop-end nucleotides in the full Rfam database (in 26,644 hairpins from all seed sequences from Rfam 8.1)

Loop	Stem Closing Pair				
End	AU	UA	CG	GC	All
AA	318	302	2173	1098	4054
AC	94	25	293	147	628
AG	113	32	694	114	1013
AU	110	66	454	208	859
CA	671	1269	865	163	3007
CC	301	72	128	133	692
CG	42	146	1099	86	1405
CU	115	104	678	175	1133
GA	175	182	1387	2270	4202
GC	62	43	170	92	378
GG	94	235	285	160	844
GU	48	34	123	153	410
UA	359	131	450	332	1318
UC	174	257	238	324	1104
UG	65	23	1158	219	1495
UU	207	140	1204	2459	4102
All	2948	3061	11399	8133	26644

Table 2. Base-2 log-likelihood ratios using raw data from Table 1 (corrected for background frequencies of A, C, G, and U)

Loop	Stem Closing Pair				
End	AU	UA	CG	GC	All
AA	0.16	0.03	0.98	0.48	0.65
AC	-0.93	-2.89	-1.24	-1.75	-1.36
AG	-0.88	-2.76	-0.22	-2.33	-0.89
AU	-1.15	-1.94	-1.06	-1.70	-1.36
CA	1.91	2.77	0.32	-1.60	0.90
CC	1.43	-0.69	-1.76	-1.22	-0.55
CG	-1.64	0.11	1.12	-2.07	0.25
CU	-0.41	-0.61	0.19	-1.27	-0.29
GA	-0.25	-0.25	0.78	1.98	1.16
GC	-1.07	-1.66	-1.57	-1.97	-1.64
GG	-0.69	0.57	-1.04	-1.39	-0.70
GU	-1.90	-2.45	-2.49	-1.69	-1.98
UA	0.55	-0.96	-1.07	-1.02	-0.75
UC	0.18	0.69	-1.32	-0.38	-0.33
UG	-1.46	-3.01	0.75	-1.17	-0.11
UU	-0.02	-0.64	0.57	2.09	1.11

Table 3. Ratios of E-values using stem closing pair specific priors to E-values using standard priors on the full set (seed plus those found by search) of sequences in 14 Rfam families

RF	Family Properties		E-value Ratios		
Acc.	Length	Number	Mean	Max	Min
00032	26	1046	1.64	2.20	1.02
00037	28	318	1.91	2.25	1.58
00453	33	30	2.67	3.60	1.81
00196	35	8	1.21	1.83	0.75
00180	36	30	1.82	3.01	1.08
00469	36	344	0.24	0.34	0.16
00385	41	41	1.66	2.42	1.09
00496	42	13	0.86	0.97	0.75
00164	42	302	1.32	1.91	0.87
00207	44	6	1.41	2.20	0.86
00617	45	426	1.47	2.43	1.16
00197	45	25	0.99	1.13	0.87
00500	45	5	1.58	2.63	0.66
00522	46	63	1.63	2.91	0.94
Mean			1.46		

6 Conclusions

Laboratory studies indicate that there is a significant effect on RNA hairpin stability of the specific nucleotides at the interface between stem and loop. Covariance models as currently used for database non-coding RNA gene search can not capture the thermodynamic regularities know from these laboratory studies. Ideally, modification of the covariance-model-based search algorithms to jointly model the probabilities of the four nucleotides at the interface would solve this problem, but at the expense of significant programming effort. However, some of the benefits of joint modeling can be had by tricking the existing algorithms by using a P-type node for the loop ends and using a new set of priors for these nodes than depend on the consensus closing pair.

Limited testing on the fourteen shortest Rfam families with a hairpin and without a pseudoknot show that specificity does seem to improve given fixed sensitivity when this P-node trick is employed.

Additional testing is needed to be more conclusive. In order to make this feasible, a more automated way to generate parameter files for Infernal needs to be developed (currently, it involves manual cut and paste and running a side program). Also, access to a computer cluster is needed to calculate E-values for many more and much longer sequences. These tasks are currently being undertaken by the author.

Acknowledgments. This material is based upon work supported by the National Institute of General Medical Sciences under grant NIH R15GM087646. The content is solely the responsibility of the author and does not necessarily represent the official views of the National Institute of General Medical Sciences or the National Institutes of Health.

References

1. Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G.: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, Cambridge (1998)
2. Eddy, S.R., Durbin, R.: RNA Sequence Analysis Using Covariance Models. Nucleic Acids Research 22, 2079–2088 (1995)
3. Karplus, K., Barrett, C., Hughey, R.: Hidden Markov Models for Detecting Remote Protein Homologies. Bioinformatics 14, 846–856 (1998)
4. Eddy, S.R.: Hidden Markov Models. Curr. Opp. Structural Biology 6, 361–365 (1996)
5. Griffiths-Jones, S., Moxon, S., Marshall, M., Khanna, A., Eddy, S.R., Bateman, A.: Rfam: Annotating Non-coding RNAs in Complete Genomes. Nucleic Acids Research 33, D121–D124 (2005)
6. Rfam, R.N.A.: Families Database of Alignments and Covariance Models, version 9.1 (2008), http://rfam.janelia.org
7. Sjölander, K., Karplus, K., Brown, M., Hughey, R., Krogh, A., et al.: Dirichlet Mixtures: a Method for Improving Detection of Weak but Significant Protein Structure Momology. Comp. Appl. Biosci. 12, 327–345 (1996)
8. Eddy, S.R.: Infernal User's Guide, version 1.0.2 (2010), http://infernal.rfam.org
9. Nawrocki, E., Eddy, S.R.: Query-dependent Banding (QDB) for Faster RNA Similarity Searches. PLoS Comp. Bio. 3, 540–554 (2007)
10. Zucher, M.: Computer Prediction of RNA Structure. Methods Enzymology 180, 262–288 (1989)
11. Wiese, K.C., Hendricks, A.: A Hybrid Clustering/Evolutionary Algorithm for RNA Folding. In: Symp. Comp. Intelligence Bioinformatics Comp. Biol., pp. 15–21. IEEE Press, New York (2008)
12. Smith, J.A., Wiese, K.C.: Integrating Thermodynamic and Observed-Frequency Data for Non-coding RNA Gene Search. In: Priami, C., Dressler, F., Akan, O., Ngom, A. (eds.) Trans. Computational Systems Biology X, pp. 124–142. Springer, Berlin (2008)
13. Vecenie, C., Serra, M.: Stability of RNA Hairpin Loops Closed by AU Base Pairs. Biochemistry 43, 11813–11817 (2004)
14. Dale, T., Smith, R., Serra, M.: A Test of the Model to Predict Unusually Stable RNA Hairpin Loop Stability. RNA 6, 608–615 (2000)
15. Serra, M., Little, M., Axenson, T., Schadt, C., Turner, D.: RNA Hairpin Loop Stability Depends on Closing Base Pair. Nucleic Acids Research 21, 3845–3849 (1993)
16. Serra, M., Axenson, T., Turner, D.: A Model for the Stabilities of RNA Hairpins Based on a Study of the Sequence Dependence of Stability for Hairpins with Six Nucleotides. Biochemistry 33, 14289–14296 (1994)
17. Giese, R., Beschart, K., Dale, T., Riley, C., Rowan, C., Sprouse, K., Serra, M.: Stability of RNA Hairpins Closed by Wobble Base Pairs. Biochemistry 37, 1094–1100 (1998)
18. Freier, S., Kierzek, R., Jaeger, J., Sugimoto, N., Caruthers, M., Neilson, T., Turner, D.: Improved Free-Energy Parameters for Predictions of RNA Duplex Stability. Proc. Natl. Acad. Sci. USA 83, 9373–9377 (1986)
19. Staple, D., Butcher, S.: Pseudoknots: RNA Structures with Diverse Functions. PLoS Bio. 3, 956–959 (2005)
20. Chomsky, N.: Three Models for the Description of Language. IRE Trans. Information Theory 2, 113–124 (1956)
21. Gumbel, J.: Statistics of Extremes. Columbia University Press, New York (1958)

Structured Output Prediction of Anti-cancer Drug Activity

Hongyu Su, Markus Heinonen, and Juho Rousu

Department of Computer Science
P.O. Box 68, 00014 University of Helsinki, Finland
{hongyu.su,markus.heinonen,juho.rousu}@cs.helsinki.fi
http://www.cs.helsinki.fi/group/sysfys

Abstract. We present a structured output prediction approach for classifying potential anti-cancer drugs. Our QSAR model takes as input a description of a molecule and predicts the activity against a set of cancer cell lines in one shot. Statistical dependencies between the cell lines are encoded by a Markov network that has cell lines as nodes and edges represent similarity according to an auxiliary dataset. Molecules are represented via kernels based on molecular graphs. Margin-based learning is applied to separate correct multilabels from incorrect ones. The performance of the multilabel classification method is shown in our experiments with NCI-Cancer data containing the cancer inhibition potential of drug-like molecules against 59 cancer cell lines. In the experiments, our method outperforms the state-of-the-art SVM method.

1 Introduction

Machine learning has become increasingly important in drug discovery where viable molecular structures are searched or designed for therapeutic efficacy. In particular, Quantitative Structure-Activity Relationship (QSAR) models, relating the molecular structures to bioactivity (therapeutical effect, side-effects, toxicity, etc.) are routinely built using state-of-the-art machine learning methods. In particular, the costly pre-clinical *in vitro* and *in vivo* testing of drug candidates can be focused to the most promising molecules, if accurate *in silico* models are available [16].

Molecular classification—the task of predicing the presence or absense of the bioactivity of interest—has been tackled with a variety of methods, including inductive logic programming [9] and artificial neural networks [1]. During the last decade kernel methods [11,16,4] have emerged as an computationally effective way to handle the non-linear properties of chemicals. In numerous studies, SVM-based methods have obtained promising results [3,16,20]. However, classification methods focusing on a single target variable are probably not optimally suited to drug screening applications where large number of target cell lines are to be handled.

In this paper we propose, to our knowledge, the first multilabel learning approach for molecular classification. Our method belongs to the structured output

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 38–49, 2010.

prediction family [15,17,12,13], where graphical models and kernels have been successfully married in recent years. In our approach, the drug targets (cancer cell lines) are organized in a Markov network, drug molecules are represented by kernels and discriminative max-margin training is used to learn the parameters. Alternatively, our method can be interpreted as a form of multitask learning [5] where the Markov network couples the tasks (cell lines) and joint features are learned for pairs of similar tasks.

2 Methods

2.1 Structured Output Learning with MMCRF

The model used is this paper is an instantiation of the structured output prediction framework MMCRF [13] for associative Markov networks and can also be seen as a sibling method to HM^3[12], which is designed for hierarchies. We give a brief outline here, the interested reader may check the details from the above references.

The MMCRF learning algorithm takes as input a matrix $K = (k(x_i, x_j))_{i,j=1}^m$ of kernel values $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ between the training patterns, where $\phi(x)$ denotes a feature description of an input pattern (in our case a potential drug molecule), and a label matrix $Y = (\mathbf{y}_i)_{i=1}^m$ containing the multilabels $\mathbf{y}_i = (y_1, \ldots, y_k)$ of the training patterns. The components $y_j \in \{-1, +1\}$ of the multilabel are called microlabels and in our case correspond to different cancer cell lines. In addition, the algorithm assumes an associative network $G = (V, E)$ to be given, where node $j \in V$ corresponds to the j'th component of the multilabel and the edges $e = (j, j') \in E$ correspond to a microlabel dependency structure.

The model learned by MMCRF takes the form of a conditional random field with exponential edge-potentials,

$$P(\mathbf{y}|x) \propto \prod_{e \in E} \exp\left(\mathbf{w}_e^T \varphi_e(x, \mathbf{y}_e)\right) = \exp\left(\mathbf{w}^T \varphi(x, \mathbf{y})\right),$$

where $\mathbf{y}_e = (y_j, y_{j'})$ denotes the pair of microlabels of the edge $e = (j, j')$. A joint feature map $\varphi_e(x, \mathbf{y}) = \phi(x) \otimes \psi_e(\mathbf{y}_e)$ for an edge is composed via tensor product of input $\phi(x)$ and output feature map $\psi(\mathbf{y})$, thus including all pairs of input and output features. The output feature map is composed of indicator functions $\psi_e^u(\mathbf{y}) = [\![\mathbf{y}_e = u]\!]$ where u ranges over the four possible labelings of an edge given binary node labels. The corresponding weights are denoted by $\mathbf{w}_e$. The benefit of the tensor product representation is that context (edge-labeling) sensitive weights can be learned for input features and no prior alignment of input and output features needs to be assumed.

The parameters are learned by maximizing the minimum loss-scaled margin between the correct training examples $(x_i, \mathbf{y}_i)$ and incorrect pseudo-examples

$(x_i, \mathbf{y}), \mathbf{y} \neq \mathbf{y}_i$, while controlling the norm of the weight vector. The primal soft-margin optimization problem takes the form

$$\begin{aligned} \underset{\mathbf{w},\xi\geq 0}{\text{minimize}} \; & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t. } & \mathbf{w}^T\varphi(x_i, \mathbf{y}_i) - \mathbf{w}^T\varphi(x_i, \mathbf{y}) \geq \ell(\mathbf{y}_i, \mathbf{y}) - \xi_i, \\ & \text{for all } i \text{ and } \mathbf{y}, \end{aligned} \tag{1}$$

where ξ_i denote the slacks allotted to each example. The effect of loss-scaling is to push high-loss pseudo-examples further away from the correct example than the low-loss pseudo-examples, which, intuitively, decreases the risk of incurring high-loss. We use *Hamming loss*

$$\ell_\Delta(\mathbf{y}, \mathbf{u}) = \sum_j [\![y_j \neq u_j]\!]$$

that is gradually increasing in the number of incorrect microlabels so that we can make a difference between 'nearly correct' and 'clearly incorrect' multilabel predictions.

The MMCRF algorithm [13] optimizes the model (1) in the so called marginal dual form, that has several benefits: the use of kernels to represent high-dimensional inputs, and polynomial-size of the optimization problem with respect to the size of the output structure. Efficient optimization is achieved via the conditional gradient algorithm [2] with feasible ascent directions found by loopy belief propagation over the Markov network G.

2.2 Kernels for Drug-Like Molecules

A major challenge for any statistical learning model is to define a measure of similarity. In chemical community, widely researched quantitative structure-activity relationship (QSAR) theory asserts that compounds having similar physico-chemical and geometric properties should have related bioactivity [7]. Various descriptors have been used to represent molecules with fixed-length feature vectors, such as atom counts, topological and shape indices, quantum-chemical and geometric properties [19]. Kernels computed from the structured representation of molecules extend the scope of the traditional approaches by allowing complex derived features to be used (walks, subgraphs, properties) while avoiding excessive computational cost [11].

In this paper, we experiment with a set of graph kernels designed for classification of drug-like molecules, including walk kernel [6], weighted decomposition kernel [10] and Tanimoto kernel [11]. All of them rely on representing the molecule as a labeled graph with atoms as nodes and bonds between the atoms as the edges.

Walk kernel. [8,6] computes the sum of matching walks (a sequence of labeled nodes so that there exists an edge for each pair of adjacent nodes) in a pair of graphs. The contribution of each matching walk is downscaled exponentially according to its length. We consider finite-length walk kernel where only walks of length p are counted. The finite walk kernel can be efficiently computed using dynamic programming.

Weighted decomposition kernel. [4] is an extension of the substructure kernel by weighting identical parts in a pair of graphs based on contextual information [4]. The kernel looks at matching subgraphs (*contextor*) in the neighborhood of *selector* atoms.

Tanimoto kernel. [11] is a kernel computed from two molecule fingerprints by checking the fraction of features that occur in both fingerprints of all features. *Hash fingerprints* enumerates all linear fragments of a given length, while *substructure keys* correspond to molecular substructures in a predefined set designed by domain experts. Based on good performance in preliminary studies, in this paper we concentrate on hash fingerprints.

2.3 Markov Network Generation for Cancer Cell Lines

In order to use MMCRF to classify drug molecules we need to build a Markov network for the cell lines used as the output, with nodes corresponding to cell lines and edges to potential statistical dependencies. To build the network we used auxiliary data (e.g. mRNA and protein expression, mutational status, chromosomal aberrations, DNA copy number variations, etc) available on the cancer cell lines from NCI database[1]. The basic approach is to construct from this data a correlation matrix between the pairs of cell lines and extract the Markov network from the matrix by favoring high-valued pairs. The following methods of network extraction were considered:

- Maximum weight spanning tree. Take the minimum number of edges that make a connected network whilst maximizing the edge weights.
- Correlation thresholding. Take all edges that exceed fixed threshold. This approach typically generates a general non-tree graph.

3 Experiments

3.1 NCI-Cancer Dataset

In this paper we use the NCI-Cancer dataset obtained through PubChem Bioassay[2] [18] data repository. The dataset initiated by National Cancer Institute and National Institutes of Health (NCI/NIH) contains bioactivity information of large number of molecules against several human cancer cell lines in 9 different tissue types, including leukemia, melanoma and cancers of the lung, colon, brain, ovary, breast, prostate, and kidney. For each molecule tested against a certain cell line, the dataset provide a bioactivity outcome that we use as the classes (active, inactive).

[1] `http://discover.nci.nih.gov/cellminer/home.do`

[2] `http://pubchem.ncbi.nlm.nih.gov`

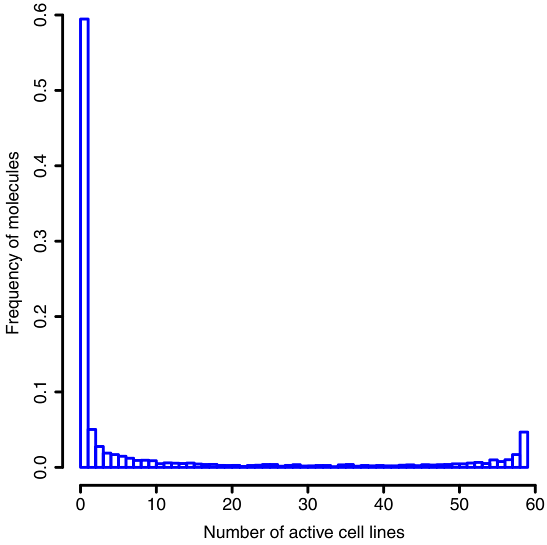


Fig. 1. Skewness of the multilabel distribution

3.2 Data Preprocessing

Currently, there are 43884 molecules in the PubChem Bioassay database together with anti-cancer activities in 73 cell lines. 59 cell lines have screen experimental results for most molecules and 4554 molecules have no missing data in these cell lines, therefore these cell lines and molecules are selected and employed in our experiments.

However, molecular activity data are highly biased over the cell lines. Figure 1 shows the molecular activity distribution over all 59 cell lines. Most of the molecules are inactive in all cell lines, while a relatively large proportion of molecules are active against almost all cell lines, which can be taken as toxics. These molecules are less likely to be potential drug candidates than the ones in the middle part of the histogram.

Figure 2 shows a heatmap of normalized Tanimoto kernel, where molecules have been sorted by the number of cell lines they are active in. The heatmap shows that the molecules in the two extremes of the multilabel distribution form groups of high similarity whereas the molecules in the middle are much more dissimilar both to each other and to the extreme groups. The result seems to indicate that the majority of molecules in the dataset are either very specific or very general in the targets they are active against. Other kernels mentioned in section 2.2 produce a similar heatmap indicating that the phenomenon is not kernel-specific.

Because of the above-mentioned skewness, we prepared different versions of the dataset:

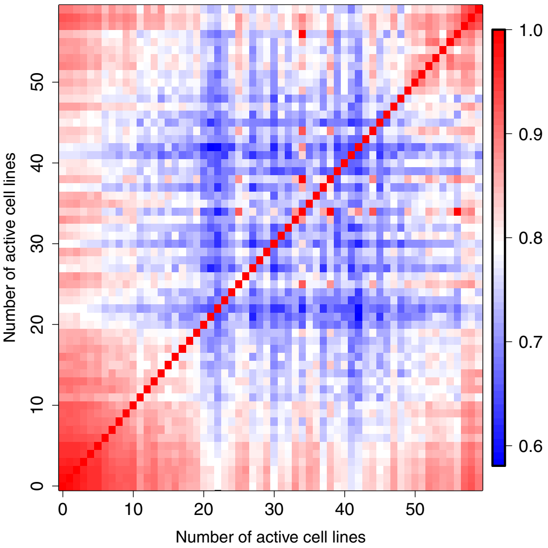


Fig. 2. Heatmap of the kernel space for the molecules sorted by the multilabel distribution

Full. This dataset contains all 4554 molecules in the NCI-Cancer dataset that have their activity class (active vs. incative) recorded against all 59 cancer cell lines.

No-Zero-Active. From this dataset, we removed all molecules that are not active towards any of the cell lines (corresponding to the leftmost peak in Figure 1). The remaining 2305 molecules are all active against at least one cell line.

Middle-Active. Here, we followed the preprocessing suggested in [14], and selected molecules that are active in more than 10 cell lines and inactive in more than 10 cell lines. As a result, 544 molecules remained and were employed in our experiments.

3.3 Experiment Setup

We conducted experiments to compare the effect of various kernels, as well as the performances of support vector machine (SVM) and MMCRF. We used the SVM implementation of the LibSVM software package written in C++[3]. We tested SVM with different margin C parameters, relative hard margin ($C = 100$) emerging as the value used in subsequent experiments. The same value was used for MMCRF classifier as well.

Because of the skewness of the multilabel distribution (c.f. 1) we used the following *stratified 5-fold cross-validation* scheme in all experiments reported:

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

we group the molecules in equivalence classes based on the number of cell lines they are active against. Then each group is randomly split among the five folds. This ensures that also the smaller groups have representation in all folds.

3.4 Kernel Setup

For the three kernel methods, walk kernel (WK) was constructed using parameters $\lambda = 0.1$ and $p = 6$ as recommended in [6]. The Weighted decomposition kernel (WDK) used context radius 3 as in [4], and a single attribute (atom type) was sufficient to give the best performance. We also used hash fragments as molecular fingerprints generated by OpenBabel[4] (using default value $n = 6$ for linear structure length), which is a chemical toolbox available in public domain. All kernels were normalized.

4 Results

4.1 Effect of Markov Network Generation Methods

We report overall prediction accuracies on the Middle-Active dataset from various Markov networks shown in Figure 3. X-axis corresponds to different microarray experiments. The accuracies from different Markov networks differ slightly. The best accuracy was achieved by using maximum weighted spanning tree approach on RNA radiation arrays dataset, shown in Figure 4, which describes profiles of radiation response in cell lines. This meets our expectations since cancer cells mostly mutated from normal cells and normal cells with radiation treatments can possibly explain the mutations.

4.2 Effect of molecule kernels

In Table 1, we report overall accuracies and microlabel F1 scores using SVM with different kernels on the Middle-Active dataset. The results were from a five-fold cross validation procedure. Here, the three kernel methods achieve almost the same accuracies in SVM classifier, while Tanimoto kernel is slightly better than others in microlabel F1 score. Thus we deemed Tanimoto kernel to be the best kernel in this experiment and chose it for the subsequent experiments.

4.3 Effect of Dataset Versions

Figure 5 gives overall accuracy and microlabel F1 score of MMCRF versus SVM for each cell line on the three versions of the data. Points above the diagonal line correspond to improvements in accuracies or F1 scores by MMCRF classifier. MMCRF improves the F1 score over SVM on each version of the data in statistically significant manner, as judged by the two-tailed sign test. Accuracy is improved in two versions, No-Zero-Actives and the Middle-Active molecules,

[4] http://openbabel.org

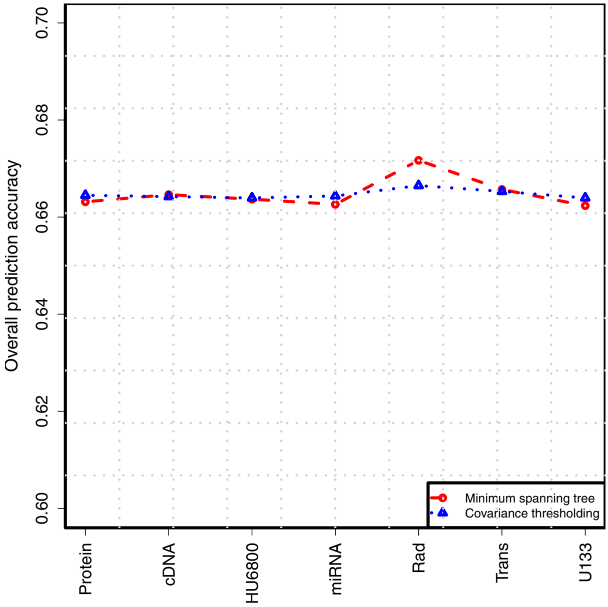


Fig. 3. Effects of Markov network construction methods and type of auxiliary data (from left to right: reverse-phase lysate arrays, cDNA arrays, Affymetric HU6800 arrays, miRNA arrays, RNA radiation arrays, transporter arrays, and Affymetrix U133 arrays)

Table 1. Accuracies and microlabel F1 scores of MMCRF and SVM with different kernels

Classifier	**Kernel**	**Accuracy**	**F1 score**
SVM	WK	64.6%	49.0%
	WDK	63.9%	51.6%
	Tanimoto	64.1%	52.7%
MMCRF	Tanimoto	67.6%	56.2%

again in statistically significant manner. Among the Middle-Active dataset, the difference in accuracy (bottom, left of Figure 5) is sometimes drastic, around 10 percentage units in favor of MMCRF for a significant fraction of the cell lines.

4.4 Agreement of MMCRF and SVM Predictions

For a closer look at the predictions of MMCRF and SVM, Table 2 depicts the agreement of the two models among positive and negative classes. Both models were trained on the Full dataset. Overall, the two models agree on the label most of the time (close to 90% of positive predictions and close to 95% of the negative predictions). MMCRF is markedly more accurate than SVM on the

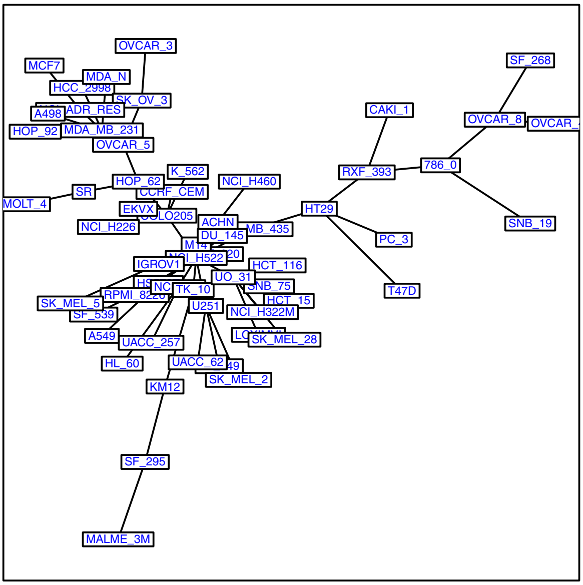


Fig. 4. Markov network constructed from maximum weighted spanning tree method on RNA radiation array data. The labels correspond to different cancer cell lines.

Table 2. Agreement of MMCRF and SVM on the positive (left) and negative (right) classes

	Positive class		**Negative class**	
	SVM Correct	SVM Incorrect	SVM Correct	SVM Incorrect
MMCRF Correct	48.6 ± 4.1%	7.1 ± 2.6%	88.0 ± 4.9%	2.2 ± 1.2%
MMCRF Incorrect	3.4 ± 1.3%	40.9 ± 3.4%	3.8 ± 1.7%	6.1 ± 3.0%

positive class while SVM is slightly more accurate among the negative class. Qualitatively similar results are obtained when the zero-active molecules are removed from the data (data not shown).

4.5 Computation Time

Besides predictive accuracy, training time of classifiers is important when a large number of drug targets need to be processed. The potential benefit of multilabel classification is the fact that only single model needs to be trained instead of a bag of binary classifiers.

We compared the running time needed to construct MMCRF classifier (implemented in native MATLAB) against libSVM classifier (C++). We conducted the experiment on a 2.0GHz computer with 8GB memory. Figure 6 shows that MMCRF scales better when training set increases.

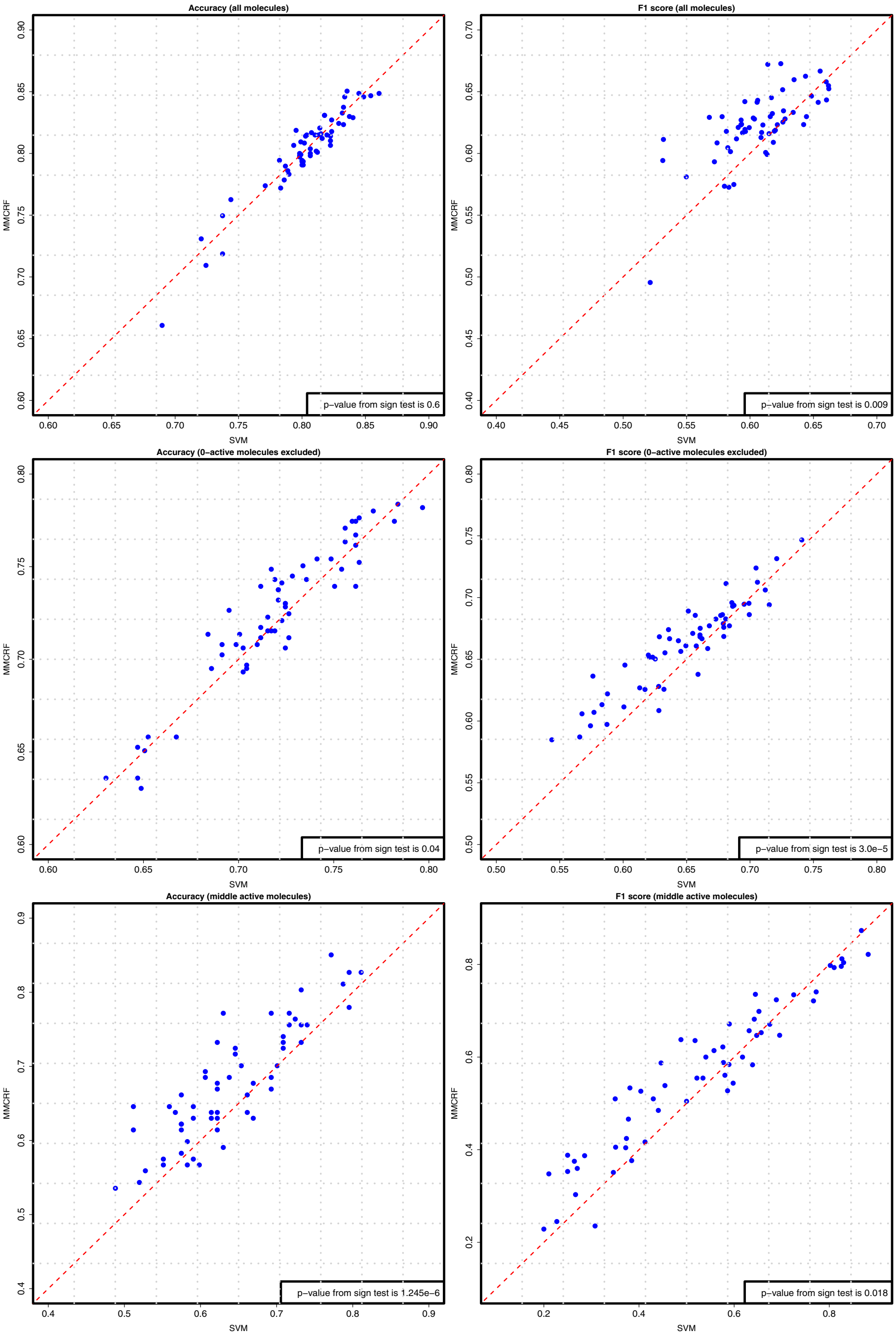


Fig. 5. Accuracy (left) and F1 score (right) of MMCRF vs. SVM on Full data (top), No-Zero-Active (middle) and Middle-Active molecules (bottom)

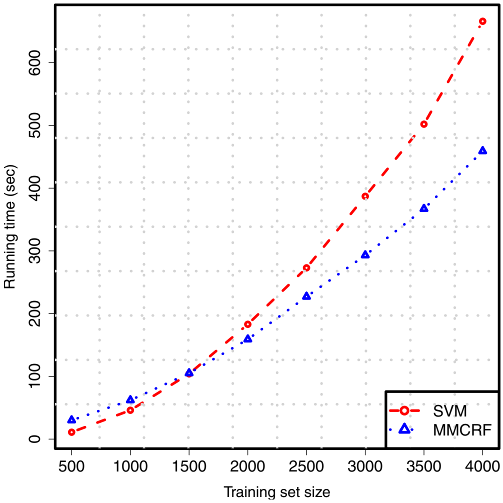


Fig. 6. Training time for SVM and MMCRF classifiers on training sets of different sizes

5 Conclusions

We presented a multilabel classification approach to drug activity classification using the Max-Margin Conditional Random Field algorithm. In the experiments against a large set of cancer lines the method significantly outperformed SVM in training time and accuracy. In particular, drastic improvements could be seen in the setup where molecules with extreme activity (active against no or a very small fraction, or a very large fraction of the cell lines) were excluded from the data. The remaining middle ground of selectively active molecules is in our view more important from drug screening applications point of view, than the two extremes.

The MMCRF software and preprocessed versions of the data are available from `http://cs.helsinki.fi/group/sysfys/software`.

Acknowledgements

This work was financially supported by Academy of Finland grant 118653 (ALGODAN) and in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, ICT-2007-216886. This publication only reflects the authors' views.

References

1. Bernazzani, L., Duce, C., Micheli, A., Mollica, V., Sperduti, A., Starita, A., Tine, M.: Predicting physical-chemical properties of compounds from molecular structures by recursive neural networks. J. Chem. Inf. Model. 46, 2030–2042 (2006)
2. Bertsekas, D.: Nonlinear Programming. Athena Scientific (1999)

3. Byvatov, E., Fechner, U., Sadowski, J., Schneider, G.: Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. J. Chem. Inf. Comput. Sci. 43, 1882–1889 (2003)
4. Ceroni, A., Costa, F., Frasconi, P.: Classification of small molecules by two- and three-dimensional decomposition kernels. Bioinformatics 23, 2038–2045 (2007)
5. Evgeniou, T., Pontil, M.: Regularized multi–task learning. In: KDD'04, pp. 109–117. ACM Press, New York (2004)
6. Gärtner, T.: A survey of kernels for structured data. SIGKDD Explor. Newsl. 5(1), 49–58 (2003)
7. Karelson, M.: Molecular Descriptors in QSAR/QSPR. Wiley-Interscience, Hoboken (2000)
8. Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized kernels between labeled graphs. In: Proceedings of the 20th International Conference on Machine Learning (ICML), Washington, DC, United States (2003)
9. King, R., Muggleton, S., Srinivasan, A., Sternberg, M.: Structure-activity relationships derived by machine learning: the use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. PNAS 93, 438–442 (1996)
10. Menchetti, S., Costa, F., Frasconi, P.: Weighted decomposition kernels. In: International Conference on Machine Learning, pp. 585–592. ACM Press, New York (2005)
11. Ralaivola, L., Swamidass, S., Saigo, H., Baldi, P.: Graph kernels for chemical informatics. Neural Networks 18, 1093–1110 (2005)
12. Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylor, J.: Kernel-Based Learning of Hierarchical Multilabel Classification Models. JMLR 7, 1601–1626 (2006)
13. Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylor, J.: Efficient algorithms for max-margin structured classification. Predicting Structured Data, 105–129 (2007)
14. Shivakumar, P., Krauthammer, M.: Structural similarity assessment for drug sensitivity prediction in cancer. Bioinformatics 10, S17 (2009)
15. Taskar, B., Guestrin, C., Koller, D.: Max-margin markov networks. In: Neural Information Processing Systems 2003 (2003)
16. Trotter, M., Buxton, M., Holden, S.: Drug design by machine learning: support vector machines for pharmaceutical data analysis. Comp. and Chem. 26, 1–20 (2001)
17. Tsochantaridis, I., Hofmann, T., Joachims, T., Altun, Y.: Support vector machine learning for interdependent and structured output spaces. In: ICML'04, pp. 823–830 (2004)
18. Wang, Y., Bolton, E., Dracheva, S., Karapetyan, K., Shoemaker, B., Suzek, T., Wang, J., Xiao, J., Zhang, J., Bryant, S.: An overview of the pubchem bioassay resource. Nucleic Acids Research 38, D255–D266 (2009)
19. Xue, Y., Li, Z., Yap, C., et al.: Effect of molecular descriptor feature selection in support vector machine classification of pharmacokinetic and toxicological properties of chemical agents. J. Chem. Inf. Comput. Sci. 44, 1630–1638 (2004)
20. Zernov, V., Balakin, K., Ivaschenko, A., Savchuk, N., Pletnev, I.: Drug discovery using support vector machines. The case studies of drug-likeness, agrochemical-likeness, and enzyme inhibition predictions. J. Chem. Inf. Comput. Sci. 43, 2048–2056 (2003)

SLiMSearch: A Webserver for Finding Novel Occurrences of Short Linear Motifs in Proteins, Incorporating Sequence Context

Norman E. Davey[1], Niall J. Haslam[2],
Denis C. Shields[2], and Richard J. Edwards[3]

[1] Structural and Computational Biology Unit,
European Molecular Biology Laboratory, 69117 Heidelberg, Germany
[2] School of Medicine and Medical Sciences, UCD Complex and Adaptive Systems Laboratory & UCD Conway Institute of Biomolecular and Biomedical Sciences,
University College Dublin, Dublin, Ireland
[3] School of Biological Sciences, University of Southampton, Southampton, UK
davey@embl.de,
{niall.haslam,denis.shields}@ucd.ie,
r.edwards@southampton.ac.uk

Abstract. Short, linear motifs (SLiMs) play a critical role in many biological processes. The SLiMSearch (Short, Linear Motif Search) webserver is a flexible tool that enables researchers to identify novel occurrences of pre-defined SLiMs in sets of proteins. Numerous masking options give the user great control over the contextual information to be included in the analyses, including evolutionary filtering and protein structural disorder. User-friendly output and visualizations of motif context allow the user to quickly gain insight into the validity of a putatively functional motif occurrence. Users can search motifs against the human proteome, or submit their own datasets of UniProt proteins, in which case motif support within the dataset is statistically assessed for over- and under-representation, accounting for evolutionary relationships between input proteins. SLiMSearch is freely available as open source Python modules and all webserver results are available for download. The SLiMSearch server is available at: http://bioware.ucd.ie/slimsearch.html.

Keywords: short linear motif, motif discovery, minimotif, elm.

1 Introduction

The purpose of the SLiMSearch (Short, Linear Motif Search) webserver is to allow researchers to identify novel occurrences of pre-defined Short Linear Motifs (SLiMs) in a set of sequences. SLiMs, also referred to as linear motifs or minimotifs, are functional microdomains that play a central role in many diverse biological pathways [1]. SLiM-mediated biological processes include post-translational modification (including cleavage), subcellular localization, and ligand binding [2]. SLiMs are typically less than ten amino acids long and have less than five defined positions, many of which will be "degenerate" and incorporate some degree of flexibility in

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 50–61, 2010.

terms of the amino acid at that position. Their length and degeneracy gives them an evolutionary plasticity which is unavailable to domains meaning that they will often evolve convergently, adding new functionality to proteins [1]. SLiMs hold great promise as future therapeutic targets, which makes their discovery of great interest [3-4].

Once a SLiM has been defined, finding matches in a given set of protein sequences is a fairly trivial task. Finding biological motifs is a standard pattern recognition task in bioinformatics. Several web-based methods to discover novel instances of known SLiMs are available, including ELM [2], MnM [5], SIRW [6] ScanProsite [7] and QuasiMotifFinder [8], which generally utilize databases of known motif patterns to search query protein sequences supplied by the user. Whilst finding matches is trivial, however, interpreting their biological significance is far from easy. The small, degenerate nature of SLiMs makes stochastic occurrences of motifs common; distinguishing real occurrences from the background of random motif hits remains the greatest challenge in *a priori* motif discovery. One approach is to simply filter out motifs that are likely to occur numerous times by chance – ScanProsite [7], for example, has an option to "Exclude motifs with a high probability of occurrence", while QuasiMotifFinder [8] uses the background occurrence of motifs in PfamA families [9] to assess the significance of hits. These strategies work well for longer, family descriptor motifs (such as are found in the Prosite database [10] used by both ScanProsite and QuasiMotifFinder) but are not so useful for SLiMs because of their tendency to occur by chance. Instead, additional contextual information such as sequence conservation [5, 8, 11-12], structural context [5, 13] or even biological keywords [6] can be used to assess the likelihood of true functional significance for putatively functional sites.

Most motif search tools rely on pre-existing motif libraries, such as ELM [2], MnM [5] or Prosite [10]. Those that permit users to define their own motifs, such as ScanProsite [7], are generally lacking the contextual information required to aid functional inference. Recent developments in *de novo* motif discovery has given rise to a number of tools that are capable of predicting entirely novel SLiMs from sets of protein sequences (*e.g.* PRATT [14], MEME [15], Dilimot [16], SLiMDisc [17] and SLiMFinder [18]). Although SLiMFinder [18] estimates the statistical significance of returned motif predictions, correcting for biases introduced by evolutionary relationships within the data, assessing the *biological* significance of predicted SLiMs remains challenging. On approach is to compare candidate SLiMs to existing motif libraries to identify similarities to previously known motifs [19].When a genuinely novel motif is predicted, however, knowledge of existing motifs is of limited use. Instead, it is useful to be able to establish the background distribution of occurrences of the novel motif, utilizing contextual information to help screen out the inevitable spurious chance matches.

We recently made our powerful *de novo* SLiM discovery tool, SLiMFinder [18], available as a webserver [20]. To aid interpretation of SLiMFinder results, we have made a new tool available, SLiMSearch, which allows users to search protein datasets with user-defined motifs, including motif prediction output from SLiMFinder. SLiMSearch utilizes the same sequence context assessment as SLiMFinder, enabling results to be masked or ranked based on the important biological indicators of sequence conservation and structural disorder [12, 21]. SLiMSearch also features the same SLiMChance algorithm for assessing statistical over-representation of SLiM occurrences, correcting for biases introduced by evolutionary relationships within the data. SLiMSearch is open source and freely available for download. For ease of use, the main SLiMSearch features have been made available as a webserver, which enables the user to search proteins for occurrences of user-specified motifs. Motifs can be searched against small custom datasets of proteins from UniProt [22]. Alternatively, searches can be performed against the whole human proteome, or defined subsets of it. Underlying methods, results formats and visualizations are fully compatible with our existing SLiM analysis webservers, SLiMDisc [23], CompariMotif [19] and SLiMFinder [20], providing a suite of integrated tools for analyzing these biologically important sequence features.

2 The SLiMSearch Algorithm

SLiMSearch performs its motif finding in three phases: (1) Input sequences are read and masked; (2) Motifs are searched against masked sequences using standard regular expression searches; (3) Motif statistics are calculated for identified motif occurrences. If desired, input sequences, input motifs and motif occurrences can be filtered based on attributes such as length, number of positions, motif conservation *etc.* SLiMs have a tendency to occur in disordered regions of proteins [24] and IUPred [21] protein disorder predictions can be used for input masking or ranking/filtering results as described further below. Conservation scoring uses the Relative Local Conservation (RLC) score introduced by Davey *et al.* [12] as implemented in SLiMFinder [20]. Conservation scoring can use pre-generated alignments or construct alignments of predicted orthology using GOPHER [23], which estimates evolutionary relationships using BLAST [25] to identify the closest-related orthologue in each species in the chosen search database. Each putative orthologue retained is: (a) more closely related to the query than any other protein from the same species; (b) related to the query through a predicted speciation event, not a duplication event.

2.1 SLiMChance Calculations of Significance

SLiMSearch utilizes a variation of the SLiMChance algorithm from SLiMFinder [18], which is based on the binomial statistics introduced by ASSET [26] and calculates the *a priori* probability of observing each motif in each sequence using the (masked) amino acid frequencies of input sequences. Observed support is then compared to expectation at two levels: (1) the total number of occurrences in all sequences; (2) the number of individual sequences returning the motif. This enables different questions to be asked of different data types. SLiMChance has an important extension over the statistics used by ASSET, and homologous proteins are optionally weighted (as in SLiMDisc [17] and SLiMFinder [18]) to account for the dependencies introduced into the probabilistic framework by homologous proteins; in this case, SLiMSearch will also assess these weighted support values. Whereas SLiMFinder is explicitly using *over*-representation to identify motifs, it is also of potential interest to see if a given motif has been avoided in a given dataset and is *under*-represented versus random expectation. The SLiMSearch implementation of SLiMChance therefore features an additional extension where the cumulative binomial probability is used to estimate the probability of seeing by chance the observed support *or less* in addition to the observed support *or more*.

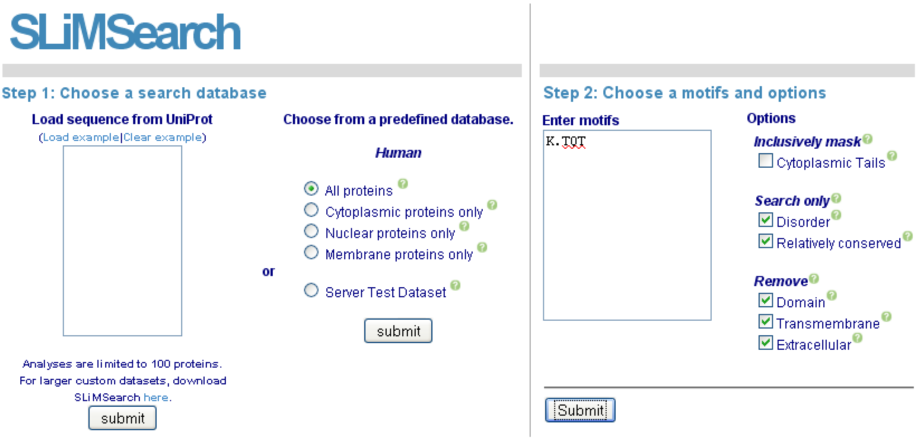
3 The SLiMSearch Webserver

The SLiMSearch server is available at: http://bioware.ucd.ie/slimsearch.html. The purpose of the webserver is to allow researchers to identify novel occurrences of predefined Short Linear Motifs (SLiMs) in a set of protein sequences. Sequences are first masked according to user specifications before motif occurrences are identified using standard regular expression searches. The SLiMChance algorithm then estimates statistical significance of over- or under-representation of each motif searched. In addition to summary results for each motif, interactive output permits easy exploration and visualization of individual motif occurrences. The context of each SLiM occurrence is then calculated in terms of protein disorder and evolutionary conservation to help the user gain insight into the validity of a putatively functional motif occurrence. The webserver is powered by the same code as the standalone version of SLiMSearch, which can be downloaded from the server. The main features of the webserver are described in more detail in the following sections.

3.1 Input

As input, SLiMSearch needs a set of protein sequences and a set of motif definitions, which are selected by the user in turn (Fig. 1). Whereas the standalone SLiMSearch program allows searching of any protein sequences, the webserver restricts the user to using UniProt sequences [22]. This is because the server relies on pre-computed alignments to keep run times down. Using UniProt downloads also allows all the masking options to be utilized (*e.g.* sequence features). The user is presented with a choice of two main input types (Fig. 1): (1) a chosen set of up to 100 UniProt entries can be downloaded for analysis; (2) the user can select from a series of predefined protein datasets. Currently, the human proteome from SwissProt [22] is available, along with three subsets defined by their subcellular localization annotation: cytoplasmic proteins, nuclear proteins and transmembrane proteins. Future server releases will expand this to other species. When searching these large proteome datasets, the evolutionary filtering [18] is switched off. To search different datasets, including datasets over 100 proteins with evolutionary filtering, users are encouraged to download and install a local version of SLiMSearch.

Once a dataset has been selected, the user must input a set of motifs to search (Fig. 1). The SLiMSearch server takes a list of motifs, typed or pasted directly into the text box. Motifs themselves are constructed from a number of regular expression elements, which are mostly standard but with a couple of additional elements to represent “3of5” motifs [27] (Table 1). SLiMSearch accepts the same input formats as CompariMotif [19], including a plain list of regular expressions and output from SLiMDisc [23] or SLiMFinder [20]. Because the focus of SLiMSearch is *short* linear motifs, the maximum number of consecutive wildcards allowed by the server is nine. Motifs must have at least *two* defined (*i.e.* non-wildcard) positions.



SLiMSearch

Step 1: Choose a search database

Load sequence from UniProt
(Load example|Clear example)

Analyses are limited to 100 proteins.
For larger custom datasets, download
SLiMSearch here.

submit

or

Choose from a predefined database.

Human

● All proteins
○ Cytoplasmic proteins only
○ Nuclear proteins only
○ Membrane proteins only

○ Server Test Dataset

submit

Step 2: Choose a motifs and options

Enter motifs

K.TQT

Options

Inclusively mask
□ Cytoplasmic Tails

Search only
☑ Disorder
☑ Relatively conserved

Remove
☑ Domain
☑ Transmembrane
☑ Extracellular

Submit

Fig. 1. SLiMSearch input options pages. Users must first either select a predefined human protein dataset, or enter a list of up to 100 UniProt IDs for a custom dataset. Clicking "submit" will then progress to Step 2, in which users enter a list of motifs for searching and set any masking options.

Table 1. Regular expression elements recognized by SLiMSearch

Element	Description
A	Single fixed amino acid.
[AB]	Ambiguity, **A** or **B**. Any number of options may be given, *e.g.* **[ABC]** = **A** or **B** or **C**.
<R:m:n>	At least **m** of a stretch of **n** residues must match **R**, where **R** is one of the above regular expression elements (single or ambiguity).
<R:m:n:B>	Exactly **m** of a stretch of **n** residues must match **R** and the rest must match **B**, where **R** and **B** are each one of the above regular expression elements (single or ambiguity). *E.g.* **<F:1:2:[DE]>** will match **[DE]F**, or **F[DE]**.
[^A]	Not **A**.
X or **.**	Wildcard positions (any amino acid).
.{m,n}	At least **m** and up to **n** wildcards.
R{n}	**n** repetitions of **R**, where **R** is any of the above regular expression elements.
^	Beginning of sequence
$	End of sequence
(R|S)	Match **R** or **S**, which are both themselves recognizable regular expressions. These motifs are not currently supported by the SLiMChance statistics and, as such, any motifs in this format with be first split into variants, *e.g.* **(R|S)PP** would be split into **RPP** and **SPP** and each searched separately.

3.2 Masking Options

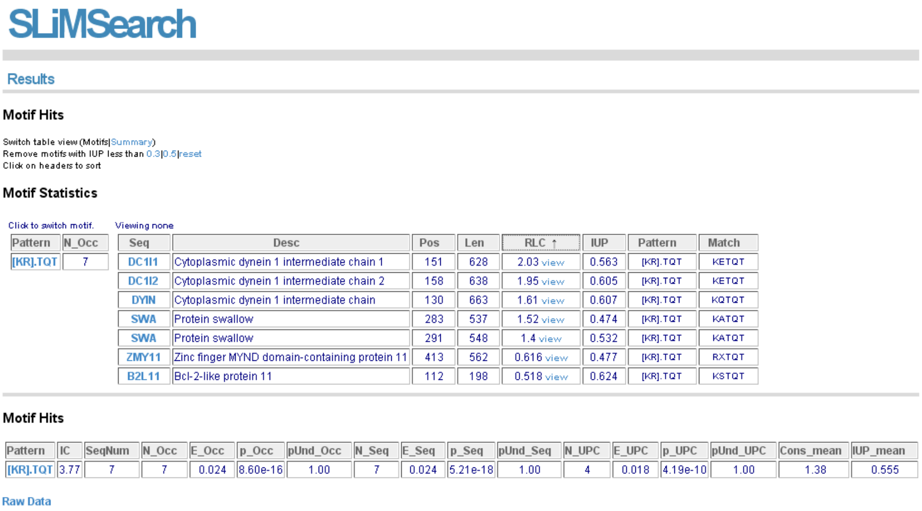
The standalone SLiMSearch program features all the input masking options of SLiMFinder [18]. For simplicity, these have been pared down for the webserver to three sets of masking options (Fig. 1): (1) restricting searches to cytoplasmic tails and loops of transmembrane proteins; (2) masking out structurally ordered regions (as predicted by IUPred [21] with a conservative threshold of 0.2) and/or relatively under-conserved residues [12]; (3) masking out domains, transmembrane and/or extracellular regions as annotated by UniProt [22]. Any combination of these options is permitted; users could, for example, restrict searches to cytoplasmic tails and loops of transmembrane proteins *and* mask out regions of predicted order, under-conserved residues and regions annotated as domains in UniProt.

3.3 Submitting Jobs

Once options have been chosen, clicking "Submit" will enter the job in the run queue. Run times will vary according to input data size and complexity, masking options and the current load of the server; the server has a maximum run time of 4 hours, after which jobs will be terminated. (For larger searches, users are encouraged to download and install a local version of SLiMSearch.) Each job is allocated a unique, randomly determined identifier. Users can either wait for their jobs to run, or bookmark the page and return to it later. Previously run job IDs can also be entered into a box on the SLiMSearch homepage to retrieve the run status and/or results.

3.4 Output

Once a job has run, the SLiMSearch results pages will open (Fig. 2). The main results page consists of a table of motif occurrences for each motif along with statistics for each occurrence including conservation (RLC) and disorder (IUPred). All fields can be sorted by clicking column headings and direct links to UniProt entries for each sequence are provided. The second primary results page consists of a summary table, which provides summary statistics for each motif. These include numbers of occurrences and SLiMChance assessments of over- or under-representation versus random expectation. Explanations of each field can be found in the SLiMSearch manual, which is available from the website. All the raw results files can also be downloaded for further analysis. When a user-defined dataset has been searched, these raw data files include the UniProt download. A key feature of SLiMSearch when analyzing user-defined datasets is the adjustment of the SLiMChance over- and under-representation statistics for evolutionary relatedness; for example, the probability of observing the Dynein Light Chain ligand "[KR].TQT" [28] in its annotated ELM proteins [2] *by chance* increases by eight orders of magnitude from 5.2e-18 to 4.2e-10 when the effective dataset size is reduced from 7 to 4 due to evolutionary relationships (Fig. 2). Whilst, in this example, the motif is still highly significant (the search dataset was defined based on the presence of the motif), in other cases this could be the difference between non-significance and apparent significance. Due to the size of the datasets, SLiMChance correction for evolutionary relationships is not available for human proteome searches.



SLiMSearch

Results

Motif Hits

Switch table view (Motifs|Summary)
Remove motifs with IUP less than 0.3|0.5|reset
Click on headers to sort

Motif Statistics

Click to switch motif.

Pattern	N_Occ
[KR].TQT	7

Viewing none

Seq	Desc	Pos	Len	RLC ↑	IUP	Pattern	Match
DC1I1	Cytoplasmic dynein 1 intermediate chain 1	151	628	2.03 view	0.563	[KR].TQT	KETQT
DC1I2	Cytoplasmic dynein 1 intermediate chain 2	158	638	1.95 view	0.605	[KR].TQT	KETQT
DYIN	Cytoplasmic dynein 1 intermediate chain	130	663	1.61 view	0.607	[KR].TQT	KQTQT
SWA	Protein swallow	283	537	1.52 view	0.474	[KR].TQT	KATQT
SWA	Protein swallow	291	548	1.4 view	0.532	[KR].TQT	KATQT
ZMY11	Zinc finger MYND domain-containing protein 11	413	562	0.616 view	0.477	[KR].TQT	RXTQT
B2L11	Bcl-2-like protein 11	112	198	0.518 view	0.624	[KR].TQT	KSTQT

Motif Hits

Pattern	IC	SeqNum	N_Occ	E_Occ	p_Occ	pUnd_Occ	N_Seq	E_Seq	p_Seq	pUnd_Seq	N_UPC	E_UPC	p_UPC	pUnd_UPC	Cons_mean	IUP_mean
[KR].TQT	3.77	7	7	0.024	8.60e-16	1.00	7	0.024	5.21e-18	1.00	4	0.018	4.19e-10	1.00	1.38	0.555

Raw Data

Fig. 2. SLiMSearch results pages. The main results page consists of a table of motif occurrences for each motif (top panel) along with statistics for each occurrence including conservation (RLC) and disorder (IUPred). All fields can be sorted by clicking column headings. Clicking sequence names will open the corresponding UniProt entry, while clicking "View" generates a visual representation of the motif. Clicking on different motifs in the smaller table on the left switches the motif being viewed. A summary table can also be viewed (bottom panel), which provides summary statistics for each motif. These statistics include SLiMChance assessments of over- or under-representation versus random expectation. Explanations of each field can be found in the SLiMSearch manual, which is available from the website. All the raw results files can also be accessed via the "Raw Data" link.

Individual motif occurrences can also be visualized for contextual information (Fig. 3). The multiple sequence alignment used for evolutionary conservation calculations is shown, with the relative conservation and IUPred disorder scores plotted below. Regions predicted to be ordered (below the disorder threshold of 0.2) are shaded, indicating areas that were (or would be) masked with disorder masking. In addition to these data, additional annotation from key SLiM and Protein databases is added. Annotated and unannotated Regular Expression matches to SLiMs from the Eukaryotic Linear Motif (ELM) database [2] are displayed above the alignment; sequence features from UniProt [22], including annotated domains and known mutations, are displayed between the alignment and RLC/Disorder plots. Users can hover the mouse over these features for additional information.

3.5 Getting Help

The SLiMSearch webserver is supported by an extensive help section, including a quickstart guide and walkthrough with screenshots. Example input files are provided. Fully interactive example output (corresponding to running the example Dynein Light Chain ligand input with default parameters) is clearly linked from the help pages. Additional details of the algorithms and options can be found in the SLiMSearch manual, which is also clearly linked from the help pages.

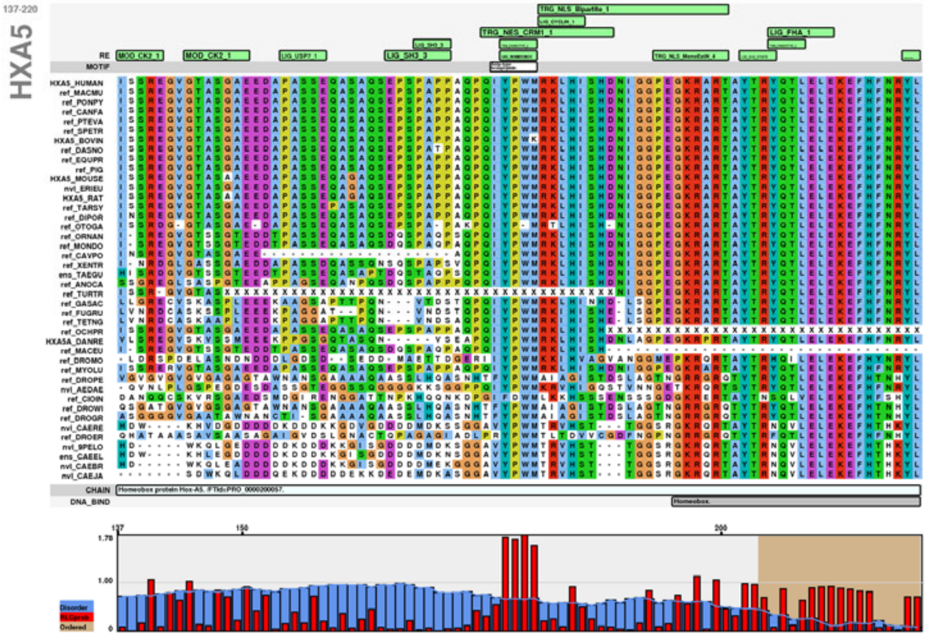


Fig. 3. Visualization of LIG_HOMEOBOX in HXA5 containing a multiple alignment of the orthologs of HXA5, drawn using Clustal coloring scheme, surrounded by relevant annotation. The bottom section contains a graph of relative conservation (in red) and IUPred disorder (in blue), with regions below the disorder threshold of 0.2 shaded (in brown). Above this section UniProt features are plotted, for example, in the case of HXA5 the right most region contains a DNA-binding Homeobox domain. Above the alignment, the motif row specifies regions containing a known functional motif (in white) and the RE row species regions matching the regular expression of a known motif (in green).

3.6 Server Limits

The server is currently limited to jobs with a run time of fewer than 4 hours. Motifs must have at least two non-wildcard positions defined and individual motif occurrence data is restricted to motifs with no more than 2000 occurrences in the search dataset. Custom UniProt datasets can have no more than 100 proteins. For larger analyses, users must install a local copy of the SLiMSearch software.

4 Example Analysis: HOX Ligand Motif

Homeobox (HOX) genes are a family of transcription factors controlling organization of segmental identity during embryo development [29] and recognized by a 60 residue DNA binding domain known as a Homeodomain [30]. HOX proteins recruit another Homeobox-containing transcription factor, PBX, via a conserved [FY][DEP]WM motif (“LIG_HOMEOBOX” [2]), binding a hydrophobic pocket created upon association of PBX to DNA [31]. Alone, the Homeodomain has weak specificity and affinity binding to the short DNA sequence TNAT, however following the formation of a heterodimer complex with TGAT binding PBX, bi-partite recognition increases specificity and allows HOX to specifically target developmental genes for expression.

A survey of the human proteome for [FY][DEP]WM PBX-binding motifs was completed to illustrate the effect of masking of globular regions and under conserved residues on the ability of a motif discovery tool to return functional motifs. Without any masking, SLiMSearch returned 53 motifs in 53 proteins, including the 16 annotated functional instances from the ELM database [2] (Supplementary Table 1). Of the 53 human occurrences, however, 30 were no longer returned following masking (IUPred masking cut-off 0.2, relative conservation filtering, domain masking and removal of extracellular and transmembrane regions). Of these 30, only 3 were known to be functional. The 23 remaining instances are all members of the Homeobox family; 13 of these contain a known annotated PBX-binding motif; given the homology of the remaining non-ELM containing proteins to the proteins containing function motifs, it is likely that all 23 instances are functional. The HXA5 occurrence, for example, shows a clear conservation signal characteristic of a functional motif despite not being annotated in ELM (Fig. 3).

5 Future Work

In addition to evolutionary conservation and structural disorder, successful identification of novel functional motifs in proteins can benefit from keyword or GO term enrichment [6, 32]. We are currently working on the incorporation of GO term enrichment into SLiMSearch analyses for future releases of the webserver. The current server is also limited to the human proteome only. In future we will expand this to include other organisms. Initially, these will be taken from the EnsEMBL database of eukaryotic genomes [33] and then expanded to other taxonomic groups [34]. We welcome suggestions from users, however, and will work with specific interest groups to add proteomes from appropriate species to the webserver where possible.

6 Conclusion

Discovering and annotating novel occurrences of Short Linear Motifs is an important ongoing task in biology, which often involves motif searches combined with additional evolutionary analyses (*e.g.* [32, 35]). The SLiMSearch webserver provides the biological community with an important advance in this arena, allowing evolutionary and structural context to be automatically incorporated into motif searches and visualized in user-friendly output. The flexibility of input, allowing known or novel motifs and user-defined protein datasets, combined with the statistical framework of SLiMChance for assessing motif abundance, makes SLiMSearch a powerful tool that should ease future discoveries of functional SLiM occurrences. In addition to the webserver implementation, SLiMSearch is available as standalone open source Python code under a GNU license, making it accessible to analyses of experimental biologists and bioinformatics specialists alike.

The SLiMSearch server is available at: http://bioware.ucd.ie/slimsearch.html. Supplementary Table 1 can be viewed at :http://bioware.ucd.ie/~compass/Server_pages/help/slimsearch/slimsearch_s1.pdf

Acknowledgments. This work was supported by Science Foundation Ireland, the University of Southampton and a European Molecular Biology Laboratory, EMBL Interdisciplinary Postdoc (EIPOD) fellowship (to N.E.D.).

References

1. Diella, F., Haslam, N., Chica, C., Budd, A., Michael, S., Brown, N.P., Trave, G., Gibson, T.J.: Understanding eukaryotic linear motifs and their role in cell signaling and regulation. Front Biosci. 13, 6580–6603 (2008)
2. Gould, C.M., Diella, F., Via, A., Puntervoll, P., Gemund, C., Chabanis-Davidson, S., Michael, S., Sayadi, A., Bryne, J.C., Chica, C., Seiler, M., Davey, N.E., Haslam, N., Weatheritt, R.J., Budd, A., Hughes, T., Pas, J., Rychlewski, L., Trave, G., Aasland, R., Helmer-Citterich, M., Linding, R., Gibson, T.J.: ELM: the status of the 2010 eukaryotic linear motif resource. Nucleic Acids Res. 38, D167–D180 (2010)
3. Kadaveru, K., Vyas, J., Schiller, M.R.: Viral infection and human disease–insights from minimotifs. Front Biosci. 13, 6455–6471 (2008)
4. Neduva, V., Russell, R.B.: Peptides mediating interaction networks: new leads at last. Curr. Opin. Biotechnol. 17, 465–471 (2006)
5. Rajasekaran, S., Balla, S., Gradie, P., Gryk, M.R., Kadaveru, K., Kundeti, V., Maciejewski, M.W., Mi, T., Rubino, N., Vyas, J., Schiller, M.R.: Minimotif miner 2nd release: a database and web system for motif search. Nucleic Acids Res. 37, D185–D190 (2009)
6. Ramu, C.: SIRW: A web server for the Simple Indexing and Retrieval System that combines sequence motif searches with keyword searches. Nucleic Acids Res. 31, 3771–3774 (2003)
7. de Castro, E., Sigrist, C.J., Gattiker, A., Bulliard, V., Langendijk-Genevaux, P.S., Gasteiger, E., Bairoch, A., Hulo, N.: ScanProsite: detection of PROSITE signature matches and ProRule-associated functional and structural residues in proteins. Nucleic Acids Res. 34, W362–W365 (2006)
8. Gutman, R., Berezin, C., Wollman, R., Rosenberg, Y., Ben-Tal, N.: QuasiMotiFinder: protein annotation by searching for evolutionarily conserved motif-like patterns. Nucleic Acids Res. 33, W255–W261 (2005)
9. Bateman, A., Coin, L., Durbin, R., Finn, R.D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E.L., Studholme, D.J., Yeats, C., Eddy, S.R.: The Pfam protein families database. Nucleic Acids Res. 32, D138–D141 (2004)
10. Sigrist, C.J., Cerutti, L., de Castro, E., Langendijk-Genevaux, P.S., Bulliard, V., Bairoch, A., Hulo, N.: PROSITE, a protein domain database for functional characterization and annotation. Nucleic Acids Res. 38, D161–D166 (2010)
11. Chica, C., Labarga, A., Gould, C.M., Lopez, R., Gibson, T.J.: A tree-based conservation scoring method for short linear motifs in multiple alignments of protein sequences. BMC Bioinformatics 9, 229 (2008)
12. Davey, N.E., Shields, D.C., Edwards, R.J.: Masking residues using context-specific evolutionary conservation significantly improves short linear motif discovery. Bioinformatics 25, 443–450 (2009)

13. Via, A., Gould, C.M., Gemund, C., Gibson, T.J., Helmer-Citterich, M.: A structure filter for the Eukaryotic Linear Motif Resource. BMC Bioinformatics 10, 351 (2009)
14. Jonassen, I., Collins, J.F., Higgins, D.G.: Finding flexible patterns in unaligned protein sequences. Protein Sci. 4, 1587–1595 (1995)
15. Bailey, T.L., Boden, M., Buske, F.A., Frith, M., Grant, C.E., Clementi, L., Ren, J., Li, W.W., Noble, W.S.: MEME SUITE: tools for motif discovery and searching. Nucleic Acids Res. 37, W202–W208 (2009)
16. Neduva, V., Russell, R.B.: DILIMOT: discovery of linear motifs in proteins. Nucleic Acids Res. 34, W350–W355 (2006)
17. Davey, N.E., Shields, D.C., Edwards, R.J.: SLiMDisc: short, linear motif discovery, correcting for common evolutionary descent. Nucleic Acids Res. 34, 3546–3554 (2006)
18. Edwards, R.J., Davey, N.E., Shields, D.C.: SLiMFinder: A Probabilistic Method for Identifying Over-Represented, Convergently Evolved, Short Linear Motifs in Proteins. PLoS ONE 2, e967 (2007)
19. Edwards, R.J., Davey, N.E., Shields, D.C.: CompariMotif: quick and easy comparisons of sequence motifs. Bioinformatics 24, 1307–1309 (2008)
20. Davey, N.E., Haslam, N.J., Shields, D.C., Edwards, R.J.: SLiMFinder: a web server to find novel, significantly over-represented, short protein motifs. Nucleic Acids Res. (2010)
21. Dosztanyi, Z., Csizmok, V., Tompa, P., Simon, I.: IUPred: web server for the prediction of intrinsically unstructured regions of proteins based on estimated energy content. Bioinformatics 21, 3433–3434 (2005)
22. Bairoch, A., Apweiler, R., Wu, C.H., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M.J., Natale, D.A., O'Donovan, C., Redaschi, N., Yeh, L.S.: The Universal Protein Resource (UniProt). Nucleic Acids Res. 33, D154–D159 (2005)
23. Davey, N.E., Edwards, R.J., Shields, D.C.: The SLiMDisc server: short, linear motif discovery in proteins. Nucleic Acids Res. 35, W455–W459 (2007)
24. Russell, R.B., Gibson, T.J.: A careful disorderliness in the proteome: sites for interaction and targets for future therapies. FEBS Lett. 582, 1271–1275 (2008)
25. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 25, 3389–3402 (1997)
26. Neuwald, A.F., Green, P.: Detecting patterns in protein sequences. J. Mol. Biol. 239, 698–712 (1994)
27. Seiler, M., Mehrle, A., Poustka, A., Wiemann, S.: The 3of5 web application for complex and comprehensive pattern matching in protein sequences. BMC Bioinformatics 7, 144 (2006)
28. Lo, K.W., Naisbitt, S., Fan, J.S., Sheng, M., Zhang, M.: The 8-kDa dynein light chain binds to its targets via a conserved (K/R)XTQT motif. J. Biol. Chem. 276, 14059–14066 (2001)
29. Wellik, D.M.: Hox genes and vertebrate axial pattern. Curr. Top Dev. Biol. 88, 257–278 (2009)
30. Gehring, W.J., Affolter, M., Burglin, T.: Homeodomain proteins. Annu. Rev. Biochem. 63, 487–526 (1994)
31. Sprules, T., Green, N., Featherstone, M., Gehring, K.: Lock and key binding of the HOX YPWM peptide to the PBX homeodomain. J. Biol. Chem. 278, 1053–1058 (2003)
32. Michael, S., Trave, G., Ramu, C., Chica, C., Gibson, T.J.: Discovery of candidate KEN-box motifs using cell cycle keyword enrichment combined with native disorder prediction and motif conservation. Bioinformatics 24, 453–457 (2008)

33. Hubbard, T.J., Aken, B.L., Ayling, S., Ballester, B., Beal, K., Bragin, E., Brent, S., Chen, Y., Clapham, P., Clarke, L., Coates, G., Fairley, S., Fitzgerald, S., Fernandez-Banet, J., Gordon, L., Graf, S., Haider, S., Hammond, M., Holland, R., Howe, K., Jenkinson, A., Johnson, N., Kahari, A., Keefe, D., Keenan, S., Kinsella, R., Kokocinski, F., Kulesha, E., Lawson, D., Longden, I., Megy, K., Meidl, P., Overduin, B., Parker, A., Pritchard, B., Rios, D., Schuster, M., Slater, G., Smedley, D., Spooner, W., Spudich, G., Trevanion, S., Vilella, A., Vogel, J., White, S., Wilder, S., Zadissa, A., Birney, E., Cunningham, F., Curwen, V., Durbin, R., Fernandez-Suarez, X.M., Herrero, J., Kasprzyk, A., Proctor, G., Smith, J., Searle, S., Flicek, P.: Ensembl 2009. Nucleic Acids Res. 37, D690–D697 (2009)
34. Kersey, P.J., Lawson, D., Birney, E., Derwent, P.S., Haimel, M., Herrero, J., Keenan, S., Kerhornou, A., Koscielny, G., Kahari, A., Kinsella, R.J., Kulesha, E., Maheswari, U., Megy, K., Nuhn, M., Proctor, G., Staines, D., Valentin, F., Vilella, A.J., Yates, A.: Ensembl Genomes: extending Ensembl across the taxonomic space. Nucleic Acids Res. 38, D563–D569 (2010)
35. Delpire, E., Gagnon, K.B.: Genome-wide analysis of SPAK/OSR1 binding motifs. Physiol Genomics 28, 223–231 (2007)

Towards 3D Modeling of Interacting TM Helix Pairs Based on Classification of Helix Pair Sequence

Witold Dyrka[1], Jean-Christophe Nebel[2], and Malgorzata Kotulska[1]

[1] Institute of Biomedical Engineering and Instrumentation, Wroclaw University of Technology, Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
[2] Faculty of Computing, Information Systems and Mathematics, Kingston University, Penhryn Road, Kingston-upon-Thames, KT1 2EE, United Kingdom
{witold.dyrka,malgorzata.kotulska}@pwr.wroc.pl, j.nebel@kingston.ac.uk

Abstract. Spatial structures of transmembrane proteins are difficult to obtain either experimentally or by computational methods. Recognition of helix-helix contacts conformations, which provide structural skeleton of many transmembrane proteins, is essential in the modeling. Majority of helix-helix interactions in transmembrane proteins can be accurately clustered into a few classes on the basis of their 3D shape. We propose a Stochastic Context Free Grammars framework, combined with evolutionary algorithm, to represent sequence level features of these classes. The descriptors were tested using independent test sets and typically achieved the areas under ROC curves 0.60-0.70; some reached 0.77.

Keywords: stochastic context-free grammar, evolutionary algorithm, helix-helix interaction, transmembrane protein.

1 Introduction

It has been estimated that around 30% of proteins in human body are transmembrane (TM) proteins [1]. Moreover, since they are more accessible to drugs than intracellular proteins, they are prime targets for drug design. Unfortunately, the specific environment of cell membranes, their large size and dynamic behavior (e.g. ion channels) make them very difficult objects for current experimental techniques in structural biology: fewer than 2% of currently known protein structures are from TM proteins [2]. Thus, the lack of experimental structures cannot be compensated by template-based modeling, i.e. homology and threading, which would require availability of a large dataset of structures. The alternative is use of ab initio methods, which build protein 3D models directly from their sequences. However, these approaches have only been successful for small proteins up to 200 amino acids [3], mainly because computational power limits the size of the conformational phase space that can be searched. Moreover, the energy function is not accurate enough to guarantee the minimum at the native state [4]. Therefore, for larger proteins, such as protein channels, which typically contain

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 62–73, 2010.

1000's of amino acids, limitations of ab initio methods can only be overcome by integrating additional knowledge in the modeling process.

Contact maps have been shown to be promising constraints. It was estimated that as few as one contact in every eight residues would be sufficient to find the correct fold of a single domain protein [5]. Moreover, even the prediction of a few contacts is useful to constrain conformational searches in ab initio prediction [6]. Recent study also suggests that some contacts are structurally more significant than others [7]. Consequently, the prediction of intramolecular contacts has become an active field of research. According to [4] homologous template approaches achieve the highest accuracy (up to 50%). However, they are not suitable for TM proteins, since very few templates are available. As correlated mutations methods have the lowest accuracy (around 20%), machine learning methods seem to be the most appropriate.

Over 80% of known TM structures are classified as alpha-helical [2]. In these proteins, molecular contacts between helices are crucial as they provide a structural skeleton. A stable interaction between two helices requires that several residues from each helix are involved in the helix-helix contact. We call this structure a helix-helix (H-H) interface and define it more precisely later in the paper. A recent study by Walters and DeGrado [8] on helix packing motifs has revealed that 90% of known configurations of H-H interactions in TM proteins can be accurately represented using only a set of 8 3D templates (Fig. 2,3 in [8]). In their research, helix pairs were clustered according to the 3D similarity (RMSD $\leq$ 1.5 A) of their fragments involved in the H-H contact. Their study also highlighted position-specific sequence propensities of amino-acids and the occurrence of the well known [GAS]-X-X-X-[GAS] motif [9].

The problem of H-H interaction prediction was addressed in [3] by creating sequence profiles from a library of helix pairs whose spatial configurations were known. In their method a helix pair in the query was compared to helix pairs in the library by calculating profile-profile scores between the pairs. While the overall accuracy of helix packing prediction was rather low, it was sufficient to constrain ab initio prediction of TM protein structures. Significantly, this approach does not model interactions between contacting residues from the two helices since this would require a more complex model than sequence profiles. Waldispuehl and Steyaert [10] proposed a multi-tape S-attributed grammar to represent helix bundles in TM proteins. In their model, a single pair of helices is described by a set of grammar rules of a non-probabilistic context-free language. At each stage of processing of a sequence, a value or attribute that reflects folding cost is calculated. The authors report that the predictive power gained from the ability to represent long range dependencies between contact residues allowed their method to outperform the best TM helix prediction software.

There are two main approaches for learning grammar rules: Maximum A Posteriori (MAP) Expectation-Maximization algorithms (EM) and evolutionary methods (Genetic Algorithms (GA) [11,12,13] or Genetic Programming (GP) [14]). Both EM and GP approaches managed to, respectively, learn probabilities of Stochastic Context-Free Grammars (SCFG) for RNA structure prediction

[15,16,17] and derive non-probabilistic CFGs for non-biological problems [18]. Successful applications of evolutionary algorithms to SCFG [19,20,21] include our earlier research on SCFGs for protein binding sites [22]. Since, unlike EM techniques, GA-based grammar inference allows introducing pressure towards more compact grammars (see Methods) and is less dependent on initial estimates of rule parameters [20], we choose this approach for learning grammar rules.

In this work we exploit the expressive power of Stochastic Context-Free Grammars to represent the subtle and complex sequence motifs underlying H-H interactions in TM proteins. The aim is to facilitate sequence based classification of helix pairs regarding their three-dimensional configuration. As a result, a class template can be assigned to a pair of helices with high accuracy. This would be extremely valuable to constrain ab initio protein structure predictions or for threading refinement.

2 Materials and Methods

2.1 Datasets

The first dataset was created on the basis of Walters and DeGrado (WDG) dataset [8]. It includes fragments of helix sequences that are in contact. We consider only the 4 most populous contact types (classes 1-4). Unlike the original set where lengths of fragments varied from 10 to 14, we kept only the 10 residues which provided the closest match with a class template. The second dataset is based on the non-redundant set of alpha-helical chains from PDBTM database [2] as of 30th November 2009. Then TM alpha helices with at least one contact residue according to Promotif3 [23] were extracted. RMSD to the representatives of the 4 WDG classes were calculated. A helix pair was assigned to a certain class if its RMSD was lower than the highest RMSD in the class of the original WDG set, i.e. 0.66, 0.93, 0.76 and 1.11A for classes 1 to 4 respectively. As a result, the PDBTM set comprises 641 helix pairs with a population of 174, 107, 64 and 69 assigned to classes 1 to 4, respectively. For training, each class used the 20 fragments which were the closest to their representative (PDBTM20). Finally, homologous sequences (40%) were removed using PAM250 matrix [24] from our combined training and test sets so that both sets were mutually independent. As result, the processed WDG test sets (WDGNR) contained 92, 49, 37 and 27 helix pair fragments for classes 1 to 4 respectively.

2.2 Principles and Formal Definitions

Amino-acid interactions between helices are subtle and complex in comparison to intra-helical interactions. Moreover, they display either parallel or anti-parallel topologies. Methods typically used for the purpose of protein pattern detection, Profile HMMs [25], cannot express these dependencies. Therefore, to classify the contact type class, we use a SCFG, which, not only, is capable of representing anti-parallel dependencies, but also can be induced automatically from a set of unrelated protein sequences which share common features [22]. The formal definition of a context-free grammar G is the following [26]: $G =< V, T, P, S >$, where V is a finite set of non-terminal (NT) symbols, T is a finite set of terminal symbols, P is a finite set of production rules and S is a special start symbol ($S \in V$). The sets V and T are mutually exclusive. Each rule from the set P has the form: $A \rightarrow X$, where $A \in V$ and $X \in (V \cup T)*$. For a SCFG, probabilities are attributed to each rule. Usually, probabilities of all productions for one Left-Hand Side (LHS) symbol sum to one; the SCFG is then called proper.

Helix interface is defined as a set of residues which are in contact with residues from the other helix, i.e. distance between residues in contact cannot be greater than the sum of van der Waals radii of their atoms enlarged by 0.6A [27]. The residues of the inner or contact face of a helix are separated by either 1 or 2 residues of the outer face so that an average helix periodicity of 3.6 residue is preserved. Two helices are separated by a coil. In the anti-parallel configuration these can be described schematically by context-free grammar rules, such as [10]:

```
Interface -> InsideRes1 Outerface InsideRes2  | Turn
Outerface -> OutsideRes1 Interface OutsideRes2 | Turn
```

More specifically, we modified a non-probabilistic CFG proposed in [10] to obtain a grammar that imposes helix periodicity (3-4 residues) and is manageable within our probabilistic scheme (i.e. not extending ca. 200 rules):

```
Start -> [ Whatever OuterfaceP Whatever }
 | [ Whatever InterfaceP Whatever }
OuterfaceP -> TwoRes InterfaceP TwoRes | OneRes InterfaceL TwoRes
 | TwoRes InterfaceR OneRes | OneRes InterfaceB OneRes | Turn
OuterfaceL -> TwoRes InterfaceP TwoRes
 | TwoRes InterfaceR OneRes | Turn
OuterfaceR -> TwoRes InterfaceP TwoRes
 | OneRes InterfaceL TwoRes | Turn
OuterfaceB -> TwoRes InterfaceP TwoRes | Coil
InterfaceP -> TwoRes OuterfaceP TwoRes | OneRes OuterfaceL TwoRes
 | TwoRes OuterfaceR OneRes | OneRes OuterfaceB OneRes | Turn
InterfaceL -> TwoRes OuterfaceP TwoRes
 | TwoRes OuterfaceR OneRes | Turn
InterfaceR -> TwoRes OuterfaceP TwoRes
 | OneRes OuterfaceL TwoRes | Turn
InterfaceB -> TwoRes OuterfaceP TwoRes | Turn
Turn -> Whatever ] { Whatever
Whatever -> X Whatever | empty
TwoRes -> OneRes OneRes
```

where the symbols '[', ']', '{' and '}' refer to the beginning and end of helix 1 and helix 2 respectively. Four *Outer-face* and *Interface* NT symbols (marked with suffixed P, L, R, B) ensure that each complete helix turn is 3 or 4 amino-acids long, e.g. if *Outer-faceP* is one-residue long, it can only be followed by *InterfaceB* which is always two-residue long. Production rule

$Turn \rightarrow Whatever]\{Whatever$ imposes helix boundaries on parser by using] and { terminal symbols. Moreover, the *Whatever* non-terminal allows to deal with parts of the helix that are not involved in the contact and thus do not share contact pattern.

2.3 Representation of Amino-Acid Properties

OneRes symbol refers to one amino-acid in a sequence. However, instead of using the amino-acid identity, which would make the grammar induction intractable, information about the level of a physio-chemical property (described later in this section) of a residue is carried. More specifically, *OneRes* can be one of three NT symbols that represent low, medium and high level of the property of interest, e.g. van der Waals volume: $OneRes \equiv Low|Medium|High$. The rationale behind this representation is to integrate quantitative information about amino-acid properties into our stochastic framework. An important advantage of this method is that it reduces the number of possible combinations of the Right-Hand Side (RHS) symbols in production rules. Therefore, a number of rules, which is maintainable in the learning process, is kept without losing generality of the grammar in the beginning of induction. For each given property, our method relies on defining all the terminal rules in the form:

```
Low     ->  amino-acid identity 1..20
Medium  ->  amino-acid identity 1..20
High    ->  amino-acid identity 1..20
```

and associating them with proper probabilities which are calculated using the known quantitative values associated to the amino acid identities. Since all terminal rules are fixed with given probabilities, unlike probabilities of all other rules, they do not need to be induced during the learning process. Moreover, to avoid trivial solutions, non-terminals which are Left-Hand Side (LHS) symbols in the terminal rules are prohibited from being LHS non-terminals of the other rules. We use the 5 categories of amino-acids from AAindex [28] as suggested in [22]: beta propensity, alpha and turn propensity, composition, physio-chemical properties and hydrophobicity.

2.4 Parsing

We use an implementation of the stochastic Earley parser [29]. In our framework Baum-Welch style Earley algorithm, where a probability for a certain node is calculated as a sum of probabilities of all sub trees, is used for training during grammar induction. This helps avoiding rapid convergence to trivial local minima in the absence of a negative training set. On the other hand, Viterbi style Earley algorithm is used for scanning, where a probability for any node in the parse tree is calculated as a maximal probability from all sub trees. According to our previous experiments, the Viterbi algorithm produces better discrimination between positive and negative samples and therefore it is more appropriate for scanning. Moreover [15,22] suggest that for a correctly induced grammar, the most likely parse tree could reflect structural features of a molecule. The output of the stochastic parser is the log probability of the couple of residues involved in a long range helix contact of a certain type, so it is a similarity measure, which estimates how the sequence of interest matches the rules associated to the interaction class.

2.5 Learning Method for Stochastic Context-Free Grammars

In order to generate interface specific descriptors using the rules described in the previous section, a training set composed of positive examples of sequence fragments containing the interface is used to infer rule weights. The general principle behind our framework is to start the learning process with the complete set of rules expressing prior knowledge of the intra-helix interaction. Then, during training, rule probabilities are inferred to express contact type specific dependencies. Although this approach leads to quite large sets of rules even for moderate alphabets, it avoids bias which would be introduced by additional constraints. In this work, induction is performed by a genetic algorithm.

Similarly to [22] in this work a single individual in GA represents a whole grammar. The genotype is coded with real numbers ($< 0, 1 >$) linked to rule probabilities. The original population of size 200 is initialized randomly and then iteratively subjected to evaluation, reproduction, genomic operators and finally succession. The objective function of the GA is defined as an arithmetic average of logs of probabilities returned by the parsing algorithm for all positive training samples. The reproduction step of the GA uses the tournament method with 2 competitors [30], which ensures that the selective pressure is held at the same level during the whole induction process. In addition, the diversity pressure is kept by using a sharing function that decreases fitness score of individuals on the basis of their similarity to other individuals in the population. The distance between individuals takes into account that probability of a rule depends not only on its own gene but also on all genes referring to rules with the same LHS non-terminal [22]. In each GA epoch (generation of individuals), only the poorer 50% of the population is substituted by new individuals to ensure the stability of the GA algorithm. Offspring are produced by averaging genetic information of two individuals with some random distortion in order to enhance exploratory capabilities of the algorithm. Subsequently, a classical one point mutation operator is used to mutate randomly chosen genes. The probabilities of crossover and mutation are 0.9 and 0.01 respectively. The algorithm stops when there is no further significant improvement in the best scores (ratio 1.001 over 100 iterations). The implementation of our grammar induction algorithm is based on M. Wall's GAlib library which provides a set of C++ genetic algorithm objects [22].

A new genotype to phenotype function $f2 = phene(gene(W \rightarrow XYZ))$ was designed to facilitate rapid convergence and enhance exploring capabilities of the genetic algorithm. Let $A \rightarrow BCD$ is a context-free rule with LHS non-terminal A, $gene(A \rightarrow BCD)$ is a real number from range 0 to 1 linked with $A \rightarrow BCD$ rule and $geneavg(A)$ is a mean value of all genes associated with rules that start with LHS non-terminal A. Then *tmpval*$(A \rightarrow BCD)$ is calculated in the following way:

```
if  gene(A->BCD)>2*geneavg(A)
then     tmpval(A->BCD)=gene(A->BCD)
else     tmpval(A->BCD)=gene(A->BCD)^10/(2*geneavg(A))^9.
```

Finally, normalization is carried out to obtain proper probabilities for each rule:

```
phene(A->BCD) = tmpval(A->BCD)/sum(XYZ){tmpval(A->XYZ)}.
```

Thus, *phene*$(A \rightarrow BCD)$ is the proper probability of the rule $A \rightarrow BCD$. The function assures that for a certain range of gene values, even small variations lead to significant changes in the phenotype. It reduces the number of active rules, since many of them have a near zero probability from the beginning of the induction. Thus, it speeds up the processing of each individual. The definition of the $f2$ function is consistent with a natural trend during grammar evolution where probabilities of unnecessary rules are reduced. This is an inherent property of proper stochastic grammars: distributions of probabilities with a small number of rules, which express well the pattern of interest, give better scores than even distributions of probabilities for all possible rules. After grammar induction, the final set of rules can be pruned to omit those which have a limited impact on the overall score of a scanned sequence.

Although genetic algorithms converge whatever their initial population [30], they may not find the global optimal solution. Therefore, for each grammar generation, we produced several grammars and selected the best one. Time needed for producing a grammar could take up to ca. 20 hours using Intel Xeon 2.4GHz quad-core processor systems at Wroclaw Centre for Networking and Supercomputing. The scanning took approximately one minute for parsing the whole test set by one grammar.

2.6 Protocol for Evaluation of Transmembrane H-H Interaction Prediction

For each of the four H-H interaction classes, 3 grammars were generated using PDBTM20 training set for each of the 6 selected amino-acid properties. The sequences of helix pair fragments from the WDGNR dataset were parsed for the four classes using all grammars. As a result, logs of probability that a sequence could have been generated by a given grammar were assigned to each H-H contact. The scores for positive and negative validation sets were analyzed by means of Receiver Operator Characteristics (ROC) methodology. The Area Under ROC Curve (AUC ROC) was used for general assessment of classifier quality and selection of the best grammar. In addition, Specificity and Sensitivity measures were calculated. Although for many applications it is desirable to maintain high Specificity or Sensitivity, we assume that the highest value of their product marks the optimal threshold for the parse score. For this threshold, Accuracy is provided.

3 Results and Discussion

3.1 Performance of Classifiers on Independent Test Set

The performance of grammar descriptors was assessed in a series of class-by-class classifications using WDGNR independent test set. On the basis of AUC ROC results for each class against other 3 classes, the properties, which lead to best scoring grammars, were selected. These were accessibility for class 1, van der Waals (vdW) volume for classes 2 and 3 and beta/turn propensity for class 4 (Tab. 1). The overall quality of classifiers measured by the Area under ROC curve varied from 0.59 for c4 to 0.68 for c2 and c3. The optimal thresholds for scores yielded in different balances between Sensitivity and Specificity. More precise evaluation of the classifiers is possible by analysis of their ROC curves (Fig. 1). There is a shift towards Sensitivity for c2 and a shift towards Specificity c3 vdW volume grammars. Typically, the relatively worst performance was obtained in classification of c1 vs. c4 or c2 vs. c3 classes. This is, however, consistent with

Table 1. H-H contact fragments classification performance using independent test set

Trained for	Using property	Tested against	AUCROC	Sensitivity	Specificity	Accuracy
c1	accessibility	c2	0.61	0.65	0.55	0.62
		c3	0.63	0.51	0.73	0.57
		c4	0.55	0.62	0.56	0.61
		c2+c3+c4	0.60	0.67	0.52	0.59
c2	van der Waals volume	c1	0.70	0.78	0.63	0.68
		c3	0.59	0.73	0.51	0.64
		c4	0.77	0.58	0.74	0.76
		c1+c3+c4	0.68	0.78	0.61	0.65
c3	van der Waals volume	c1	0.71	0.62	0.78	0.74
		c2	0.59	0.49	0.76	0.64
		c4	0.73	0.54	0.89	0.69
		c1+c2+c4	0.68	0.54	0.79	0.75
c4	beta-sheet propensity	c1	0.56	0.67	0.48	0.52
		c2	0.52	0.56	0.51	0.53
		c3	0.73	0.63	0.81	0.73
		c1+c2+c3	0.59	0.67	0.50	0.52

Table 2. Properties used by best class-by-class classifiers. Class-by-class classification of helix-helix pair contact fragments performance measured by Area and ROC curve using independent test set.

	c1		c2		c3		c4	
c1			accessibility	0.61	accessibility	0.63	frequency	0.57
c2	VdW volume	0.70			frequency	0.64	vdW volume	0.77
c3	vdW volume	0.71	vdW volume	0.59			vdW volume	0.73
c4	beta prop.	0.56	accessibility	0.59	beta prop.	0.73		

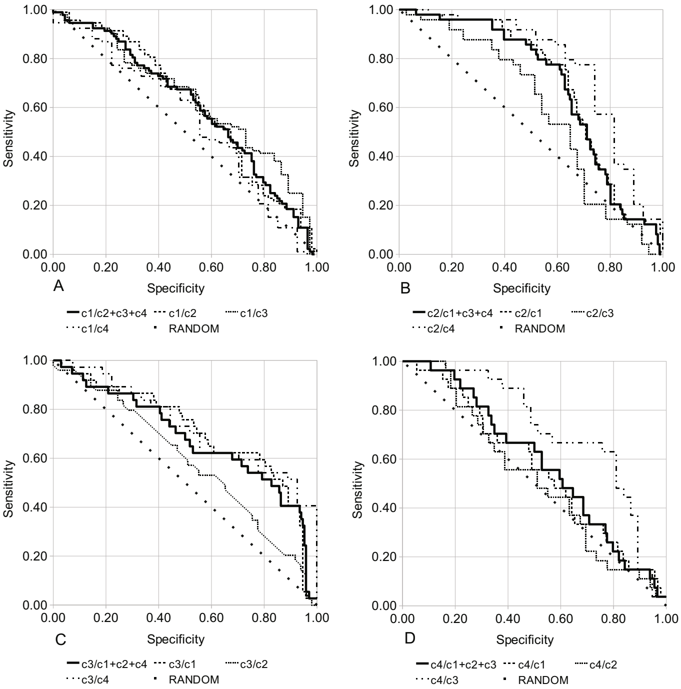


Fig. 1. ROC curves for H-H contact fragment classifiers: c1 accessibility-based (A), c2 vdW volume-based (B), c3 vdW volume-based (C) and c4 beta propensity-based (D)

the fact that these pairs of classes are similar in terms of RMSD. They differ in relative direction of helices (anti-parallel for c1 and c2, parallel for c3 and c4).

Representatives of 5 categories of amino-acid properties were utilized for grammar training resulting in varying robustness for different class-by-class comparisons. The properties that were used in best scoring grammars are presented in Table 2. In general, area under ROC curve values of the best grammars, for each class-by-class classification, were in the range from 0.56 to 0.77. Accessibility and vdW volume were most useful for distinguishing between classes unrelated in terms of their 3D shape. Frequency and beta-sheet propensity were the properties that allow for classification between anti-parallel and parallel versions of classes that share similar spatial configurations.

3.2 Analysis of Classifiers Features

Our analysis details the features of the SCFG classifiers, which contribute to the overall performance of the method. Our findings suggest that the difference in sequence composition, in terms of the property underlying the grammar, is the main factor. However, in a few cases descriptors that performed better than expected, according to sequence composition comparison, were obtained. Such examples include classifications between: c1 and c2 using grammar based on accessibility, c3 and c1+c4 using grammar based on van der Waals volume and c4

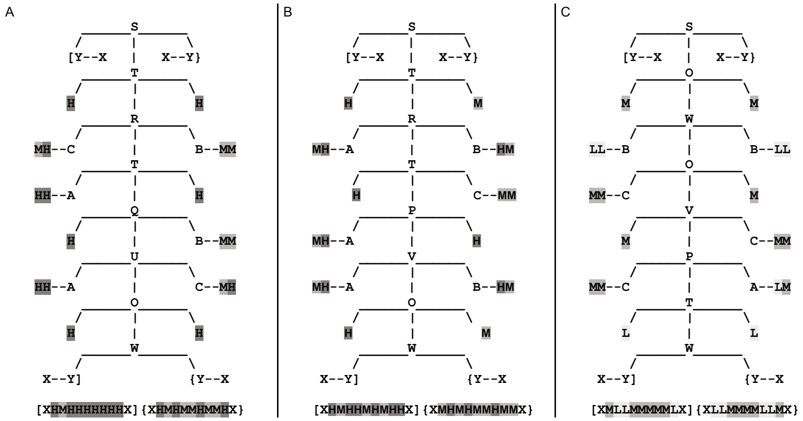

A
B
C
[XHMHHHHHHHX]{XHMHMMHMMHX}
[XHMHHMHMHHX]{XMHMHMMHMMX}
[XMLLMMMMMLX]{XLLMMMMLLMX}

Fig. 2. Parse trees that would give maximum scores for (A) c1 accessibility grammar, (B) c4 accessibility grammar and (C) c3 vdW volume grammar. H, M, L are property level NTs, which refer to high, medium, low level of a given property. X is any amino-acid (probability of 1/20 to each amino-acid type). S is a start symbol. Subsets of NTs T,U,V,W and O,P,Q,R are designated to model Inter- and Outer-face of the helix pair (order of subsets is arbitrary). The sans-serif font for property level NTs for (B) indicates a modified method of assignment of probabilities to the rules started with those symbols (in text).

and c1 using grammar based on accessibility. The last was obtained in a scheme that included modified training and test sets. Moreover, property levels were related to the average property level in a training set, instead of the average over 20 amino-acids as utilized in the basic scheme. In Fig. 2, example of parse trees that would give maximum scores for these grammars are shown. Although they would not necessarily result in maximal parse scores for individual sequences, their structure is very likely to be found in real parses. It would be difficult at this stage of study to induce relations between parse tree structures and biological features of helix pairs, especially for classes 3 and 4, which are parallel. However, the analysis of the parse trees suggests that grammar classifiers can benefit from representation of dependencies between helices. For example, in (C) the most probable rules typically require that amino-acids from two helices have similar size at each stage of derivation. These results confirm the value of a strategy which uses amino-acid properties instead of amino-acid identities for modeling non-homologous helix pair sequences. However, the exact assignment of amino-acid to property levels remains an issue. We noticed that non-terminals related to property levels underrepresented in H-H bundles were rarely used in induced grammars, which hampered the capability of representing class defining patterns.

4 Conclusions

Our SCFG framework produced sequence-based descriptors, which represent classes of transmembrane helix-helix interaction configurations. The grammar descriptors were tested using independent test sets. Amino-acid properties most relevant to each class-by-class classification were selected. Areas under ROC curves obtained for best classifiers were typically between 0.60 to 0.70 and in some cases higher. This shows that amino-acid sequence based descriptors can be used for prediction of H-H interaction structural class, for a pair of H-H sequences. Thus, they can be used to constrain the search space of an ab initio prediction method for transmembrane proteins. Another strategy could be use of predicted conformations of H-H interactions to deprive sets of structures modeled in the process of ab initio prediction of low quality items.

At this stage of research, the predictive power of the classifiers is mainly grounded in differences in amino-acid composition of H-H pairs in terms of the amino-acid properties. However, some grammar descriptors perform above expected level, based on sequence composition. This suggests that capability of CFG to represent higher level (anti-parallel) dependencies between interacting helices can contribute to the classification. Currently, we investigate the influence of several factors, including choice of the class representatives and the training sets, definition of the amino-acid property levels and design of the initial grammar structure. We also research the hypothesis that there are subclasses within WDG classes of H-H sequences more prone to structural description than others.

The other factor, important for the procedure of training, is the selection of the training set. According to recent publications [3,8], the optimal length of a helix fragment is from 10 to 14 residues. However the position of cutting of fragments could potentially have an impact on the quality of prediction. Finally, the clustering of H-H interfaces is still an open problem. The numbers of PDBTM sequences assigned to each WDG class representative were linearly correlated to the cut-off levels. This suggests, that the level of RMSD around 1.50 A prohibits the classes from overlapping but only conveys a limited biological meaning.

Acknowledgments. This work was partially supported by Ministry of Science and Higher Education of Poland (N N519 401537), British Council Young Scientists Programme (WAR/324/108) and MLODA KADRA Programme.

References

1. Yarov-Yarovoy, V., Schonbrun, J., Baker, D.: Multipass Membrane Protein Structure Prediction Using Rosetta. Proteins 62, 1010–1025 (2006)
2. Tusnady, G.E., Dosztányi, Z., Simon, I.: PDB_TM: selection and membrane localization of transmembrane proteins in the PDB. Nucleic Acids Res. 33, D275–D278 (2005)
3. Barth, P., Wallner, B., Baker, D.: Prediction of membrane protein structures with complex topologies using limited constraints. Proc. Natl. Acad. Sci. 106, 1409–1414 (2009)
4. Wu, S., Zhang, Y.: A comprehensive assessment of sequence-based and template-based methods for protein contact prediction. Bioinformatics 24, 924–931 (2008)
5. Li, W., et al.: Application of sparse NMR restraints to large-scale protein structure prediction. Biophys. J. 87, 1241–1248 (2004)
6. Izarzugaza, J.M.G., Grana, O., Tress, M.L., Valencia, A., Clarke, N.D.: Assessment of intramolecular contact predictions for CASP7. Proteins 69(suppl. 8), 152–158 (2007)

7. Sathyapriya, R., Duarte, J.M., Stehr, H., Filippis, I., Lappe, M.: Defining an Essence of Structure Determining Residue Contacts in Proteins. PLoS Comput. Biol. 5, e1000584 (2009)
8. Walters, R.F.S., De Grado, W.F.: Helix-packing motifs in membrane proteins. Proc. Natl. Acad. Sci. 103, 13658–13663
9. Russ, W.P., Engelman, D.M.: The GxxxG motif: a framework for transmembrane helix-helix association. J. Mol. Biol. 296(3), 911–919 (2000)
10. Waldispühl, J., Steyaert, J.-M.: Modeling and predicting all-transmembrane proteins including helix-helix pairing. Theoretical Computer Science 335, 67–92 (2005)
11. Holland, J.H.: Adaptation in Natural and Artificial Systems. Univ. Michigan (1975)
12. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning Reading. Addison-Wesley, Reading (1989)
13. O'Neill, M., Ryan, C.: Grammatical Evolution. IEEE Trans. Evol. Comput. 5, 349–358 (2001)
14. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
15. Sakakibara, Y., Brown, M., Underwood, R.C., Mian, I.S.: Stochastic Context-Free Grammars for Modeling RNA. In: Procs 27th Hawaii Int. Conf. System Sciences (1993)
16. Sakakibara, Y., Brown, M., Hughey, R., Mian, I.S., Sjolander, K., Underwood, R., Haussler, D.: Stochastic Context-Free Grammars for tRNA. Nucleic Acids Res 22, 5112–5120 (1994)
17. Knudsen, B., Hein, J.: RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. Bioinformatics 15, 446–454 (1999)
18. Mernik, M., Crepinsek, M., Gerlic, G., Zumer, V., Viljem, Z., Bryant, B.R., Sprague, A.: Learning CFG using an Evolutionary Approach. Technical report (2003)
19. Sakakibara, Y.: Learning context-free grammars using tabular representations. Pattern Recognition 38, 1372–1383 (2005)
20. Keller, B., Lutz, R.: Evolutionary induction of stochastic context free grammars. Pattern Recognition 38, 1393–1406 (2005)
21. Cielecki, L., Unold, O.: Real-valued GCS classifier system. Int. J. Appl. Math. Comput. Sci. 17, 539–547 (2007)
22. Dyrka, W., Nebel, J.-C.: A Stochastic Context Free Grammar based Framework for Analysis of Protein Sequences. BMC Bioinformatics 10, 323 (2009)
23. Hutchinson, E.G., Thornton, J.M.: PROMOTIF - A program to identify structural motifs in proteins. Protein Science 5, 212–220 (1996)
24. Dayhoff, M.O., Schwartz, R.M., Orcutt, B.C.: A model of evolutionary change in proteins. Atlas of Protein Sequence and Structure 5, 345–352 (1978)
25. Krogh, A., Brown, M., Mian, I.S., Sjolander, K., Haussler, D.: Hidden Markov models in computational biology: Applications to protein modeling. J. Mol. Biol. 235, 1501–1531 (1994)
26. Revesz, G.E.: Introduction to Formal Languages. McGraw-Hill, New York (1983)
27. Gimpelev, M., Forrest, L.R., Murray, D., Honig, B.: Helical Packing Patterns in Membrane and Soluble Proteins. Biophysical J. 87, 4075–4086 (2004)
28. Kawashima, S., Pokarowski, P., Pokarowska, M., Kolinski, A., Katayama, T., Kanehisa, M.: AAindex: amino acid index database. Nucleic Acids Res. 36, D202–D205 (2008)
29. Stolcke, A.: An Efficient Probabilistic Context-Free Parsing Algorithm that Computes Prefix Probabilities. Computational Linguistics 21(2), 165–201 (1995)
30. Arabas, J.: Wyklady z algorytmow ewolucyjnych Warsaw: WNT (2004)
31. Wall, M.: GAlib library documentation (version 2.4.4). MIT, Cambridge (1999)

Optimization Algorithms for Identification and Genotyping of Copy Number Polymorphisms in Human Populations

Gökhan Yavaş[1], Mehmet Koyutürk[1,3], and Thomas LaFramboise[2,3]

[1] Department of Electrical Engineering & Computer Science,
Case Western Reserve University, Cleveland, OH, USA
[2] Department of Genetics, Case Western Reserve University, Cleveland, OH, USA
[3] Center for Proteomics & Bioinformatics, Case Western Reserve University,
Cleveland, OH, USA

Abstract. Recent studies show that copy number polymorphisms (CNPs), defined as genome segments that are polymorphic with regard to genomic copy number and segregate at greater than 1% frequency in the populations, are associated with various diseases. Since rare copy number variations (CNVs) and CNPs bear different characteristics, the problem of discovering CNPs presents opportunities beyond what is available to algorithms that are designed to identify rare CNVs. We present a method for identifying and genotyping common CNPs. The proposed method, POLYGON, produces copy number genotypes of the samples at each CNP and fine-tunes its boundaries by framing CNP identification and genotyping as an optimization problem with an explicitly formulated objective function. We apply POLYGON to data from hundreds of samples and demonstrate that it significantly improves the performance of existing single-sample CNV identification methods. We also demonstrate its superior performance as compared to two other CNP identification/genotyping methods.

Keywords: CNV, CNP, optimization.

1 Introduction

Genetic differences that can be identified with single nucleotide polymorphism (SNP) microarrays include SNPs [1] and copy number variants (CNVs) [2]. CNVs are defined as chromosomal segments of at least 1000 bases (1 kb) in length that vary in number of copies from human to human. To date, several methods have been proposed for inferring CNVs from SNP array data [3-6]. In a recent study [7], we have formulated CNV identification as an optimization problem with an explicitly designed objective function that is characterized by several adjustable parameters. Our method, ÇOKGEN, efficiently identifies CNVs using a variant of the well-known simulated annealing heuristic.

All of these approaches are specifically designed for identifying rare or *de novo* CNVs by individually searching a sample's genome for regions in which evidence of copy number deviation exists. On the other hand, recent genome-wide association

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 74–85, 2010.

studies (GWAS) have underscored the importance of identifying common CNPs, associating them with several complex disease phenotypes [8-11]. Although these results highlight the need for dedicated methods for common CNP identification, most of the methods for CNV identification have not yet separated the ideas of identification and genotyping of common CNPs from discovery of rare CNVs.

In this paper, we present a method for identifying common copy number polymorphisms. The proposed method, POLYGON, takes as input the copy number variants identified by a single-sample CNV identification algorithm (e.g., ÇOKGEN [7], PennCNV [6], Birdseye [4]) and implements a computational framework to (i) identify CNVs in different samples that might correspond to the same variant in the population (candidate CNPs), (ii) adjust the boundaries of these candidate CNPs by drawing strength from raw copy number data from multiple samples, and (iii) determine copy number genotypes in the study. The key ingredient of this computational framework is an explicitly formulated objective function that takes into account several criteria, which are carefully designed to quantify the desirability of a CNP genotype with respect to various biological insights and experimental considerations. Namely, these criteria include minimizing variability in raw copy numbers of markers that are assigned to the same copy number class across samples, and maximizing raw copy number differences between samples that are assigned different copy numbers. We then develop algorithms that find copy number genotypes that optimize this function for fixed boundaries, and use this algorithm in a hierarchical manner to precisely adjust the boundaries of each CNP. Our performance analysis shows that POLYGON dramatically improves the performance of single sample methods in terms of Mendelian concordance and provides a moderate improvement in terms of sensitivity. Furthermore, we demonstrate its superior performance when compared to two other recurrent CNP detection algorithms presented in [12].

In the next section, we describe the general algorithmic framework for POLYGON, formulate CNP identification and genotyping as an optimization problem and present algorithms to solve this problem. Subsequently, in Section 3, we provide comprehensive experimental results on the performance of POLYGON in inferring CNPs from CNVs identified by three state-of-the-art CNV identification algorithms; ÇOKGEN, PennCNV, and Birdseye. We also compare the performance of our method to two other multi sample methods, COMPOSITE and COVER [12]. Finally, in Section 4, we discuss these results.

2 Methods

POLYGON first uses an existing algorithm to identify CNVs in each sample. The output of this step generates a list of CNVs for each sample, which may correspond to CNPs, rare/*de novo* CNVs, or false positives. Copy number genotypes for these CNVs are not required by POLYGON. Subsequently, POLYGON reconciles these CNVs in two phases:

(i) Clustering of identified CNVs to obtain an initial set of *candidate CNPs* (clusters of CNVs that potentially correspond to the same event).
(ii) Fine tuning of the boundaries of candidate CNPs and precise estimation of number of copies in each sample.

In the remainder of this section, we explain the algorithmic details of these two phases.

2.1 Problem Definition

Consider a study in which a set N of samples are screened via SNP microarray technology to obtain raw copy number estimates for a set M of markers on a single chromosome (we formulate the problem in the context of a single chromosome since each chromosome can be processed separately). The HapMap [1] dataset contains 270 samples and a total of approximately 1.8 million markers (for Affymetrix 6.0 SNP array) over 23 chromosomes. The objective of the CNP identification and genotyping problem is to assign a copy number to all markers in all samples such that copy number assignment is smooth across markers and consistent across samples. Formally, we are seeking a mapping S: N x M → C, where C = {0, 1, 2, 3, 4} denotes the set of possible copy numbers and 0, 1, 2, 3, and 4, respectively denote homozygous deletion, hemizygous deletion, normal copy number, hemizygous duplication, and homozygous duplication (some samples may contain more than four copies, but all such cases are encapsulated into copy number class 4 to have a compact set of copy number classes). To find the mapping, POLYGON uses two data types:

(i) The set V = $\{v_1, v_2, \ldots v_K\}$ of CNV calls provided by a single-sample algorithm. Each CNV $v \in$ V is a pair (s_v, e_v) where s_v and e_v denote the start and end markers of the region v, and $M_v=\{i\colon s_v \le i \le e_v\}$ defines the set of markers flanked by the pair. The length of CNV v is defined as $l_v = |M_v| = e_v - s_v + 1$.
(ii) For each sample marker $(n, m) \in$ N x M, the raw copy number estimate $R_{n,m}$. These estimates are also provided by the single-sample algorithms which are utilized for CNV identification.

POLYGON implements a two-phase algorithm to call CNPs from these raw copy numbers and initial set of CNVs. The aim of the first phase is to obtain a set, W $=\{w_1, w_2, .., w_t\}$, of candidate CNPs by clustering CNVs identified on different samples according to their chromosomal coordinates. Each candidate $w \in$ W is defined by the pair (s_w, e_w) where s_w and e_w represent the start and end markers of the region. Similar to M_v, $M_w=\{i\colon s_v \le i \le e_v\}$ defines the set of markers in CNP w. Based on the definition of w, we reduce the CNP genotyping problem to finding a set of functions S_w: N → C for all $w \in$ W where S_w determines the genotype of each sample at CNP w. Then, for each $(n, m) \in$ N x M**,** $S(n, m)$ is defined as $S_w(n)$ if $m \in M_w$ and 2 otherwise for all $w \in$ W.

Thus, in the second phase, we utilize an optimization based framework to find the optimal S_w for each $w \in$ W (hence we obtain the optimal genotyping of all CNPs which implies optimal S), while fine-tuning its boundaries.

2.2 Identification of Candidate CNPs

In the first phase, POLYGON clusters individual CNVs based on the start and end markers to obtain the candidate CNPs that represent "similar" CNVs on different samples. To assess the similarity between two CNVs, we use the *minimum reciprocal overlap* (*MRO*) measure. For two CNVs v_1 and v_2, let $o(v_1,v_2)=|M_{v_1} \cap M_{v_2}|$ denote the

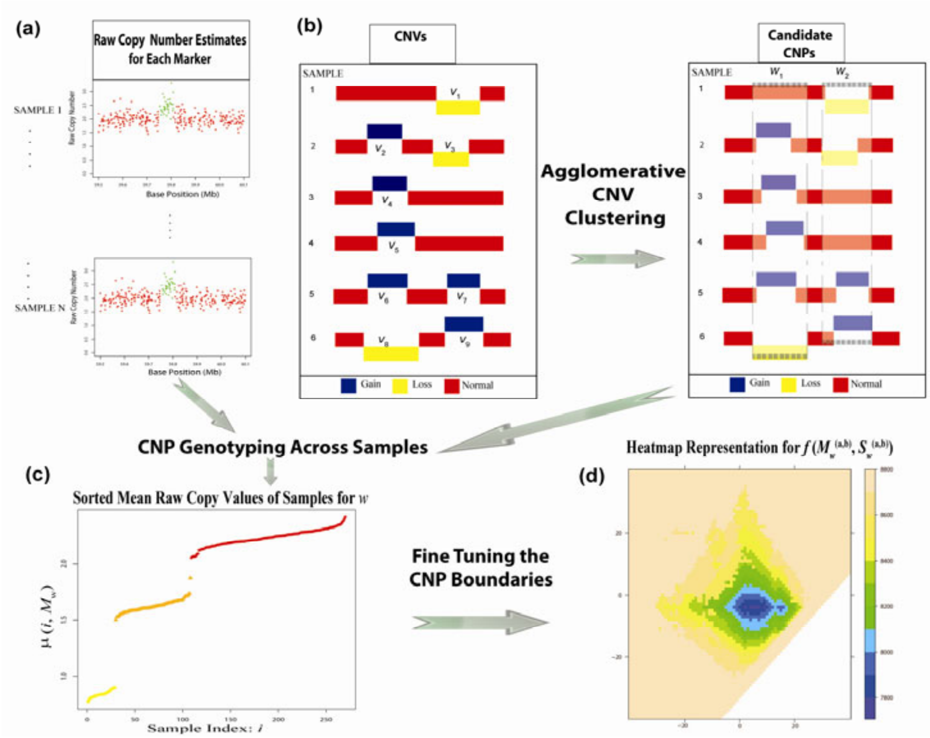

(a)
Raw Copy Number Estimates for Each Marker
SAMPLE 1
SAMPLE N
Raw Copy Number
Base Position (Mb)
(b)
CNVs
SAMPLE
v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8
v_9
Gain
Loss
Normal
Agglomerative CNV Clustering
Candidate CNPs
w_1
w_2
CNP Genotyping Across Samples
(c)
Sorted Mean Raw Copy Values of Samples for w
μ (i, M_w)
Sample Index: i
Fine Tuning the CNP Boundaries
(d)
Heatmap Representation for f(M_w^(a,b), S_w^(a,b))

Fig. 1. Algorithmic workflow of POLYGON. (a) The raw copy estimates as provided by the single-sample CNV detection algorithms. (b) Our agglomerative CNV clustering algorithm takes as input the CNVs identified by the single-sample CNV detection algorithms, to obtain a set of candidate CNPs. Here, the algorithm is illustrated on a toy example set of CNVs, V = $\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$, obtaining the set of candidate CNPs W=$\{w_1, w_2\}$. (c) For each $w \in$ W, to obtain the optimal copy number genotyping in each sample for given candidate boundaries of *w,* the samples are sorted with respect to average copy number within these boundaries. Subsequently, high gradient points in this ordering are identified to segregate samples into copy number classes. The sorted mean raw copy numbers and the associated genotypes are for a real *w* identified by POLYGON in the HapMap dataset, and are not related to the toy example of (b). The samples genotyped with copy number classes 0, 1 and 2 are shown with colors yellow, orange and red, respectively. (d) The heat map displays the matrix colored according to the values of the objective function $f(M_{\mathrm{w}}^{(\mathrm{a,b})}, S_w^{(\mathrm{a,b})})$ at the optimal genotype solution for each candidate boundary (a,b) as computed by the procedure in (c). Note that the coordinates on the horizontal and vertical axis correspond to the start and end coordinates of candidate boundaries for *w*, and that for demonstration purposes they have been re-centered so that the initial boundaries are at (0,0). Once this heatmap is obtained, the optimal boundaries of the CNP are set to (a, b) that correspond to the minimum value in this matrix and the copy number genotypes are given by the optimal assignment for those boundaries (as computed in (c)).

size of the overlap between v_1 and v_2. Then the minimum reciprocal overlap of v_1 and v_2 is defined as

$$MRO\ (v_1, v_2) = \min\left(\frac{o(v_1, v_2)}{l_{v_1}}, \frac{o(v_1, v_2)}{l_{v_2}}\right).$$

Using this similarity measure, POLYGON agglomeratively clusters CNVs using a conservative complete-linkage based criterion to measure the similarity between groups of CNVs. We use $\Pi = \{\rho_1, \rho_2, ..., \rho_t\}$ to denote a set of CNV clusters where each $\rho_i \in \Pi$ represents a set of CNVs. At the beginning of clustering, each CNV constitutes a cluster by itself, i.e., $\Pi^{(0)} = \{\{v_i\}: v_i \in \mathbf{V}\}$. At each iteration, two candidate CNV clusters with maximum similarity are merged, where the similarity between CNV clusters ρ_i and ρ_j is defined as

$$MRO\ (\rho_i, \rho_j) = \min_{\mathrm{v_q} \in \rho_\mathrm{i}, \mathrm{v_p} \in \rho_\mathrm{j}} \{MRO\ (v_q, v_p)\}\ .$$

This process continues until the similarity between any two clusters goes below a predefined threshold. The set obtained through the clustering process $\Pi = \{\rho_1, \rho_2, ..., \rho_t\}$ is then used to obtain the candidate CNP set $\mathbf{W} = \{w_1, w_2, .., w_t\}$, where each $w_i = (s_{w_i}, e_{w_i})$ and $s_{w_i} = \min_{v \in \rho_i}\{s_v\}$ and $e_{w_i} = \max_{v \in \rho_i}\{e_v\}$. In this study, we have chosen the overlap threshold as 0.5, which guarantees that all the CNVs that correspond to a single candidate CNP have at least 50% mutual overlap in terms of markers that they span. Note that we do not take into consideration the type of the CNV (*e.g.,* deletion *vs.* insertion) while clustering CNVs. Therefore, it is possible that a loss and a gain can be represented by the same candidate CNP as long as they share at least 50% of their markers. The motivation behind this approach is that both gain and loss events were reported for the same region in different samples in previous research [13]. In Figure 1(b), this process is illustrated with a toy example.

The next phase of POLYGON processes each candidate CNP individually and determines the CNP genotype of each sample, while fine tuning its boundaries.

2.3 Identifying CNP Genotypes and Fine-Tuning of CNP Boundaries

Once the set of candidate CNPs are obtained, for each CNP region w, we select a window of markers to be searched exhaustively to fine-tune the boundaries of w. The initial boundaries of the window containing w are extended to allow consideration of the markers bordering initially identified w for enlarging, shrinking or shifting its markers. We define the search window for $w \in \mathbf{W}$ as the set of markers $\Omega_w = \{i: s_w - \lceil l_w/2 \rceil \le i \le e_w + \lceil l_w/2 \rceil\}$.

In order to assess the quality of the boundaries of a CNP and the genotype calls in each sample, we formulate an objective function that brings together multiple quantitative criteria that gauge the suitability of CNP genotype calls based on observed array intensities of all the samples. This objective function takes into account the smoothness of raw copy number estimates over contiguous markers that are declared to have identical copy numbers, as well as consistency of genotype calls of the same CNP across samples.

We define objective function $f\,(M_w, S_w)$ as a combination of the following objective criteria:

- **Variation in raw copy numbers within each copy number class should be minimized.** Ideally, the raw copy number estimates (i.e., $R_{n,m}$) for markers that are assigned identical copy numbers should be similar. For a given CNP w and copy number assignment S_w, let the set of samples assigned to class $c \in$ C be $\Psi(c) = \{n \in \mathrm{N} : S_w(n) = c\}$. The mean raw copy number for class c can be computed as follows:

$$\mu(c) = \begin{cases} \dfrac{\sum\limits_{n\in\Psi(c)} \sum\limits_{m\in\Omega_w} R_{n,m} + \sum\limits_{n\in N\setminus\Psi(c)} \sum\limits_{m\in\Omega_w\setminus M_w} R_{n,m}}{|\Omega_w||\Psi(c)| + |\Omega_w \setminus M_w||N \setminus \Psi(c)|} & \text{if c = 2} \\ \\ \dfrac{\sum\limits_{n\in\Psi(c)} \sum\limits_{m\in M_w} R_{n,m}}{\mathrm{I}\, M_w \,\mathrm{I}\, |\Psi(c)|} & \text{otherwise} \end{cases}.$$

The mean raw copy number values for aberrant copy number classes are simply calculated by averaging the raw copy estimates in region M_w across all samples genotyped with the specified copy number class. However, for the “normal” copy number class, this computation is slightly more complicated since the markers in all samples that are outside the boundaries of w also contribute to the mean of the “normal” copy number class. Then, the total intra-class variability induced by S_w is given by

$$\sigma(M_w, S_w) = \sum_{c\in C\setminus 2} \sum_{n\in\Psi(c)} \left(\sum_{m\in M_w} |R_{n,m} - \mu(c)| + \sum_{m\in\Omega_w\setminus M_w} |R_{n,m} - \mu(2)| \right) + \sum_{n\in\Psi(2)} \sum_{m\in\Omega_w} |R_{n,m} - \mu(2)|.$$

Consequently, a desirable combination of M_w and S_w is expected to minimize $\sigma(M_w, S_w)$ (subject to other constraints). Note that this formulation does not make any assumption about the expected raw copy numbers at the markers and therefore is robust to any systematic bias that might be encountered in measurement and normalization of the $R_{m,n}$.

- **Variation in raw copy numbers across different copy number classes should be maximized.** The criterion formulated above focuses on the internal variation in a copy number class. However, it is also important to accurately separate different copy number classes from each other, since the number of variants in the sample is unknown and intra-class variation can be minimized by artificially increasing the number of genotype classes across samples. For this reason, we formulate an objective criterion that penalizes excessive copy number classes. Formally, we define

$$\chi(M_w, S_w) = \sum_{c=0}^{3} 2^{\frac{1}{\mu(c+1)-\mu(c)}} I(|\Psi(c)||\Psi(c+1)| \neq 0)$$

as an objective criterion to be minimized. Here $I(.)$ denotes the indicator function (*i.e.*, it is equal to 1 if the statement being evaluated is true, and 0 otherwise). Observe that this function grows exponentially with the reciprocal of the difference between the mean raw copy numbers of markers assigned to consecutive copy number classes, and is therefore minimized when similar raw copy numbers are assigned to the same class.

- **Filtering out noise by eliminating smaller regions.** Longer CNPs indicate higher confidence as it can be statistically argued that shorter sequences of markers with deviant raw copy numbers are more likely to be observed due to noise. Thus, we explicitly consider CNP length as an additional objective criterion. We then define $\lambda(M_w)=\frac{1}{2^{l_w}}$ as an objective criterion that penalizes shorter CNPs.

- **The optimal CNP identification and genotyping problem.** We use a linear combination of the criteria above as an objective function to assess the quality of a CNP region and assignment of copy number genotypes. Namely, for a given candidate CNP *w,* an assignment of markers M_w to *w*, and assignment S_w of copy numbers to these markers in each sample is defined as

$$f(M_w,S_w)=k_\sigma\sigma(M_w,S_w)+k_\chi\chi(M_w,S_w)+k_\lambda\lambda(M_w)$$

The objective of the CNP identification and genotyping problem is to find M_w and S_w that together minimize $f(M_w, S_{w.})$. Here, the tunable coefficients $k_\sigma, k_\chi, k_\lambda$ adjust the relative importance of the objective criteria with respect to each other. In our experiments, we use a prohibitively large value for k_λ to eliminate CNP instance calls on smaller regions that are likely to be false positives. The parameters k_σ and k_χ are used to adjust the apparent trade-off between the intra-class and the inter-class variation. Without loss of generality, we require that $k_\sigma+k_\chi=1$ so that the parameters can be adjusted in an interpretable way. For our experimental evaluations reported in this paper, we use $k_\sigma=0.5$ and $k_\chi=0.5$. Note also that, for a given M_w and S_w, the computation of $f(M_w, S_w)$ requires $O(|\mathbf{N}||\Omega_w|)$ time.

2.4 Algorithms for Optimal CNP Identification and Copy Number Genotyping

We now describe the algorithm we use to find the objective function minimum, thereby solving the CNP identification and genotyping problem. A solution to a given instance of the problem is characterized by assignment of marker boundaries to the CNP (M_w) along with the copy number genotyping $S_w(n)$ for each sample $n \in$ N. Consequently, an optimal solution to the problem can be determined by finding an optimal S_w for each possible M_w and choosing the best among these solutions across all possible assignments of M_w. Since a CNP region is by definition composed of contiguous markers and the problem is defined within a fixed segment of markers Ω_w, there are $|\Omega_w|(|\Omega_w|+1)/2$ possibilities for M_w, making such an exhaustive search feasible. Motivated by this insight, we now discuss how an optimal assignment of S_w can be found for fixed M_w.

(i) Optimal CNP genotyping for fixed CNP boundaries. When the boundaries of the CNP are fixed, the solution to the CNP genotyping problem is uniquely determined by the assignment of each sample to a copy number class for the CNP region at hand. To find an optimal solution to this problem, POLYGON uses a top-down approach that starts from a conservative solution that assigns all samples to the same class and iteratively improves this solution by dividing samples into separate classes as necessary. Initially, all samples are assigned to the "normal" class, *i.e.*,

$S_w^{(0)}(n) = 2$ for all $n \in$ N. At each step of the algorithm, samples that are assigned to the same copy number class are iteratively considered to check whether it is possible to further improve the solution by dividing this partition of samples into two sub partitions with different copy number classes. To find the best possible partitioning of the samples in a group, we use the mean raw copy number of markers within M_w on each sample, computed as:

$$\mu(n, M_w) = \frac{\sum_{m \in M_w} R_{n,m}}{l_w} \quad .$$

Assume, without loss of generality, that the samples are ordered according to $\mu(n, M_w)$. That is, $\mu(n, M_w) \leq \mu(n+1, M_w)$ for all $i = 1, \ldots, |\mathrm{N}|-1$. The aim of our algorithm is to divide the ordered of set samples in up to five partitions such that each partition corresponds to the set of samples with a copy number class and the objective function f is minimized for the given class assignments. It can be shown that the optimal copy number genotype assignment must preserve the $\mu(n, M_\mathrm{w})$ ordering. Based on this observation, we develop a heuristic based on the notion that a sample at which the copy number genotype change is most likely to happen is the one at which the maximum increase is observed in between $\mu(n, M_\mathrm{w})$ and $\mu(n+1, M_\mathrm{w})$ values.

Our algorithm is executed using a series of splits dividing one copy number class into two at each stage. Let $S_w^{(i)}$ denote the solution after the i^{th} split where $0 \leq i \leq 4$ (since there can be at most 5 copy number class partitions) and $\Psi^{(i)}(c)$ denotes the set of samples in the partition for copy number class $c \in$ C after the i^{th} split. In each round, our algorithm introduces a new copy number class partition by *splitting* an already existing copy number class partition c. This is done by choosing a sample n^*, and then either moving all samples $n \leq n^*$ in n^*'s copy number class c to copy number class c-1, or moving all samples $n > n^*$ in n^*'s copy number class to copy number class c+1. We call n^* a split sample. However, if the algorithm tries to split a copy number class partition by re-introducing an already existing copy number class partition (*i.e.,* if copy number c-1 or c+1 is already assigned to some samples), this split becomes invalid and our algorithm tries another n^* for this round of split procedure. Let $\mathrm{Q}^{(i)}$ denote the set of candidate split samples, *i.e*., samples that are not used in one of the previous splits or are skipped by the algorithm . Initially, we have $S_\mathrm{w}^{(0)}(n) = 2$ for all $n \in$ N , $\Psi^{(0)}(2) =$ N and $\Psi^{(0)}(c) = \varnothing$ for $c \in$ C \ 2, and $\mathrm{Q}^{(0)} =$ N.

For each sample $1 \leq n \leq |\mathbf{N}|-1$, let $\Delta(n) = \mu(n+1, M_\mathrm{w}) - \mu(n, M_\mathrm{w})$ denote the gradient of mean copy numbers at sample n. At each round of the algorithm, the sample $n^* = \mathrm{argmax}_{n \in \mathrm{Q}^{(i)}}\{\Delta(n)\}$ is selected as the splitting sample, since it would yield the highest inter-class variance for the new class partitions being created. Assume that n^* is assigned copy number c at this point. One of the sub-partitions that can be obtained by splitting the partition c will obviously be the old partition c. In order to determine whether the other sub-partition will be c-1 or c+1, we check the similarity of the mean raw copy number of each sub-partition to that of the original partition. To do so, the mean raw copy number for each sub-partition is computed as:

$$\mu_c = \frac{\sum_{j \in \Psi^{(i)}(c)} \mu(j, M_w)}{|\Psi^{(i)}(c)|} \;,\; \mu_c' = \frac{\sum_{j \in \Psi^{(i)}(c), j \leq n^*} \mu(j, M_w)}{n^* - \min(\Psi^{(i)}(c)) + 1} \;,\; \mu_c'' = \frac{\sum_{j \in \Psi^{(i)}(c), j > n^*} \mu(j, M_w)}{\max(\Psi^{(i)}(c)) - n^*} \;.$$

There are two cases to be considered.

Case 1: $\left|\mu_c - \mu_c'\right| \le \left|\mu_c - \mu_c''\right|$

In this case, the samples in the lower sub-partition have more similar mean raw copy number to that of the samples in the original partition. Therefore, the newly introduced copy number class partition should be c+1 and the samples from min($\Psi(c)$) to n^* will remain in partition c and samples from n^*+1 to max($\Psi(c)$) will be assigned to partition c+1 in the new solution, i.e.,

$$S_w^{(i+1)}(n) = \begin{cases} c+1 & \text{for } n \in \Psi^{(i)}(c) \text{ and } \ n > n^* \\ S_w^{(i)}(n) & \text{otherwise} \end{cases}.$$

Case 2: $\left|\mu_c - \mu_c'\right| > \left|\mu_c - \mu_c''\right|$

In this case, the upper sub-partition is more similar to the original partition in terms of mean raw copy number. Thus, the newly introduced copy number class partition should be c-1 and the samples from n^*+1 to max($\Psi(c)$) will be assigned to class c and samples from min($\Psi(c)$) to n^* will be assigned to class c-1 in the new solution, i.e.,

$$S_w^{(i+1)}(n) = \begin{cases} c-1 & \text{for } n \in \Psi^{(i)}(c) \text{ and } \ n \le n^* \\ S_w^{(i)}(n) & \text{otherwise} \end{cases}.$$

Note that splits in cases 1 and 2 are invalid if $\Psi^{(i)}(c+1) \neq \varnothing$ and $\Psi^{(i)}(c-1) \neq \varnothing$, respectively (i.e., the split is trying to introduce a copy number class partition that already exists). In that case, the algorithm updates the set of candidate split samples as $Q^{(i)} = Q^{(i)} \setminus n^*$, and repeats the procedure for finding a split sample for the current $S_w^{(i)}$ as described above. In the case of a valid split, it checks whether the new solution $S_w^{(i+1)}$ improves the current solution $S_w^{(i)}$ in terms of the objective function (i.e., if $f(M_w, S_w^{(i+1)}) < f(M_w, S_w^{(i)})$). If so, the algorithm sets $Q^{(i+1)} = Q^{(i)} \setminus n^*$, updates $\Psi^{(i+1)}$ according to $S_w^{(i+1)}$, and moves to the next splitting round. The algorithm will stop if the number of copy number class partitions reaches five, the set of candidate split samples becomes empty (i.e., $Q^{(i)} = \varnothing$), or the new solution $S_w^{(i+1)}$ does not improve the current solution $S_w^{(i)}$ in terms of the objective function. In these cases, $S_w^{(i)}$ is reported as the optimal solution. Note that the running time of this algorithm is $O(|N||\Omega_w|)$, since the dominant computation throughout the course of the algorithm is the computation of f for a constant number of times.

In Figure 1(c), for a CNP w, the ordered samples and the corresponding mean raw copy numbers $\mu(n, M_w)$ for each sample $n \in \{1, 2,.., 270\}$ are shown. As evident in the plot, the top candidate split samples are those where the biggest jumps occur between consecutive μ values. After applying the above procedure, we find that the CNP w manifests itself in three different copy number classes across the sample set **N**. The samples genotyped with copy number 0, 1 and 2 classes are colored with yellow, orange and red, respectively.

(ii) Finding the optimal boundaries of a candidate CNP. The above procedure gives a solution to the optimal CNP assignment problem for fixed CNP boundaries (M_w). Recall that for each CNP w, an initial estimate of its boundaries is available from the first phase of POLYGON. We exhaustively search all possible sub-windows

$[a, b]$ within Ω_w (where $s_w - \lceil l_w/2 \rceil \leq a \leq b \leq e_w + \lceil l_w/2 \rceil$), finding the optimal CNP genotyping $S_w^{(a,b)}$ for each candidate boundary $M_w^{(a,b)}$. Finally, the $M_w^{(a,b)}$ and $S_w^{(a,b)}$ that minimize $f(M_w^{(a,b)}, S_w^{(a,b)})$ are returned as the optimal CNP assignment for CNP w. This procedure is illustrated in Figure 1(d). From the heat map in the figure, it can be observed that the optimal boundaries obtained after this method is applied are different from the initial boundaries of w. The total runtime of this algorithm is $O(|N||\Omega_w|^3)$, which is reasonable in practical cases since the size of region Ω_w does not exceed several hundred markers for majority of the CNPs discovered by our method.

3 Results

We apply our algorithm to Affymetrix 6.0 SNP array data from 270 HapMap individuals. We use three different algorithms, ÇOKGEN [7], PennCNV [6] and Birdseye [4] to detect the initial set of CNVs that serve as input to POLYGON.

3.1 Methods Used for Comparison

There are few CNP identification methods available for SNP array platforms. Here we compare POLYGON with two methods, COMPOSITE and COVER, which were published quite recently [12]. Similar to POLYGON, these two methods use CNVs identified by other methods to call common CNPs. Thus, they utilize the same type of data (CNVs mined on the Affymetrix 6.0 SNP array by PennCNV and an annotation file containing the genomic coordinates of the markers) and produce the same type of output with POLYGON. It should be noted that there exists another method, Canary [4], for genotyping CNPs. However, it is designed to genotype the CNP maps given by [13] and is not a CNP discovery method *per se*. For this reason, we do not include Canary in our comparisons.

To simplify the discordance and sensitivity analysis and to be consistent with the results of the single-sample based CNV identification algorithms, a CNP genotyped by POLYGON, COMPOSITE or COVER is treated as a single gain or loss CNV event in the analyses reported here. For the discordance and sensitivity analysis, we use the *MRO* measure (as defined in Section 2.2) with a threshold of 0.5 to decide whether two CNVs identified in two different individuals correspond to the same event.

3.2 Trio Discordance Comparison across Methods

The 60 mother-father-child trios in the HapMap data set were used to assess the accuracy of CNV genotyping algorithms by measuring the rate of Mendelian concordance. A gain or loss in a trio child is said to be Mendelian concordant if it appears in at least one of the parents. Unless the CNV is *de novo*, any discordance is either the result of a false positive call in the child or a false negative call in one of the parents.

For all of the single-sample CNV identification methods, POLYGON greatly improves trio discordance. POLYGON reduces ÇOKGEN's trio discordance from 30.8% to 20.1%. Similarly, it reduces PennCNV's trio discordance from 32.9% to 16.2%. On the other hand, both COMPOSITE and COVER reduce PennCNV's trio discordance rate to around 26%. These results demonstrate the superior ability of POLYGON for CNP identification and copy number genotyping across samples.

3.3 Sensitivity Comparison across Methods

A recent study [14] assembled a “stringent dataset”, which contains CNVs identified by at least two independent algorithms. The data set contains a total of 808 autosomal CNV regions reported to be harbored in at least one of the 270 HapMap individuals. We use this as a “gold standard” data set on which to evaluate the sensitivity of our method.

POLYGON improves the sensitivity of two single-sample based CNV identification methods. While ÇOKGEN achieves a sensitivity of 86%, POLYGON improves this to 88.3%. Similarly, sensitivity increases from 84.7% to 89.9% when POLYGON is run with CNVs obtained by Birdseye. Interestingly, on the other hand, PennCNV and POLYGON on PennCNV achieve the same sensitivity rate of 88.6%. These figures are clearly superior to the sensitivity of both COMPOSITE (62.8%) and COVER (40.2%).

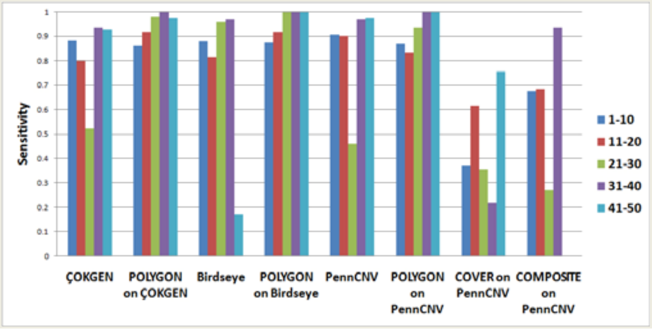


Fig. 2. Sensitivity of different algorithms. Each bar represents the sensitivity of the associated method in the specified frequency stratum.

In Figure 2, we compare the sensitivity of the methods stratified by the gain/loss frequencies of the CNVs. The purpose of this analysis is to see whether an algorithm that explicitly targets common CNPs is more successful in calling common CNPs accurately (as compared to rare CNVs). Indeed, as seen in the figure, POLYGON improves the sensitivity of all CNV identification methods for gains/losses existing in more than 20 samples, demonstrating that POLYGON is well-suited to detect common CNPs. Furthermore, for gains/losses that occur in at least 30 samples, POLYGON consistently achieves sensitivity above 98%, regardless of the algorithm that is used to identify the initial set of CNVs. This observation suggests that POLYGON is also quite robust against changes in the input set of CNVs.

4 Conclusion

We have presented a method to detect and genotype germline copy number polymorphisms (CNPs) from SNP array data and a set of CNVs. Our approach will be useful for researchers querying constitutional DNA for association of CNP alleles with disease. Indeed, CNPs are emerging as important factors in a growing number of diseases. POLYGON's ability to identify recurrent variants is particularly crucial in GWAS, as variations frequently observed in a significant proportion of the population may have a significant impact on human disease.

The current work shows that the problem of detecting CNPs may be recast as an optimization problem with an explicit objective function. The objective function chosen here is quite simple and intuitive, but its effectiveness is clear. With detailed experimental studies on the HapMap dataset, we have demonstrated its sensitivity to identify especially common CNPs, while keeping a low false positive rate, as demonstrated by high Mendelian consistency in trios.

References

1. IHMC: A haplotype map of the human genome. Nature 437, 1241–1242 (2005)
2. Feuk, L., et al.: Structural variation in the human genome. Nat. Rev. Genet. 7, 85–97 (2006)
3. Colella, S., et al.: QuantiSNP: an Objective Bayes Hidden-Markov Model to detect and accurately map copy number variation using SNP genotyping data. Nucleic Acids Res. 35, 2013–2025 (2007)
4. Korn, J.M., et al.: Integrated genotype calling and association analysis of SNPs, common copy number polymorphisms and rare CNVs. Nat. Genet. 40, 1253–1260 (2008)
5. Olshen, A.B., et al.: Circular binary segmentation for the analysis of array-based DNA copy number data. Biostatistics 5, 557–572 (2004)
6. Wang, K., et al.: PennCNV: an integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. Genome Res. 17, 1665–1674 (2007)
7. Yavaş, G., et al.: An optimization framework for unsupervised identification of rare copy number variation from SNP array data. Genome Biology 10, R119 (2009)
8. Gonzalez, E., et al.: The influence of CCL3L1 gene-containing segmental duplications on HIV-1/AIDS susceptibility. Science 307, 1434–1440 (2005)
9. Aitman, T.J., et al.: Copy number polymorphism in Fcgr3 predisposes to glomerulonephritis in rats and humans. Nature 439, 851–855 (2006)
10. Fanciulli, M., et al.: FCGR3B copy number variation is associated with susceptibility to systemic, but not organ-specific, autoimmunity. Nat. Genet. 39, 721–723 (2007)
11. Yang, Y., et al.: Gene copy-number variation and associated polymorphisms of complement component C4 in human systemic lupus erythematosus (SLE): low copy number is a risk factor for and high copy number is a protective factor against SLE susceptibility in European Americans. Am. J. Hum. Genet. 80, 1037–1054 (2007)
12. Shu Mei, T., et al.: Identification of recurrent regions of copy-number variants across multiple individuals. BMC Bioinformatics 11, 147 (2010)
13. McCarroll, S.A., et al.: Integrated detection and population-genetic analysis of SNPs and copy number variation. Nat. Genet. 40, 1166–1174 (2008)
14. Pinto, D., et al.: Copy-number variation in control population cohorts. Hum. Mol. Genet. 16, R168–R173 (2007)

Preservation of Statistically Significant Patterns in Multiresolution 0-1 Data

Prem Raj Adhikari and Jaakko Hollmén

Aalto University School of Science and Technology
Department of Information and Computer Science
P.O. Box 15400, FI-00076 Aalto
Espoo, Finland
prem.adhikari@tkk.fi, jaakko.hollmen@tkk.fi

Abstract. Measurements in biology are made with high throughput and high resolution techniques often resulting in data in multiple resolutions. Currently, available standard algorithms can only handle data in one resolution. Generative models such as mixture models are often used to model such data. However, significance of the patterns generated by generative models has so far received inadequate attention. This paper analyses the statistical significance of the patterns preserved in sampling between different resolutions and when sampling from a generative model. Furthermore, we study the effect of noise on the likelihood with respect to the changing resolutions and sample size. Finite mixture of multivariate Bernoulli distribution is used to model amplification patterns in cancer in multiple resolutions. Statistically significant itemsets are identified in original data and data sampled from the generative models using randomization and their relationships are studied. The results showed that statistically significant itemsets are effectively preserved by mixture models. The preservation is more accurate in coarse resolution compared to the finer resolution. Furthermore, the effect of noise on data on higher resolution and with smaller number of sample size is higher than the data in lower resolution and with higher number of sample size.

Keywords: Multiresolution data, statistical significance, frequent itemset, mixture modelling.

1 Introduction

Biological experiments performed with high throughput and high resolutions techniques often produce data in multiple resolutions. Furthermore, International System for human Cytogenetic Nomenclature (ISCN) has defined five different resolutions of the chromosome band: 300, 400, 550, 700 and 850[1]. In other words, chromosomes are divided into 862 regions in resolution 850 (fine resolution) and 393 regions in resolution 400 (coarse resolution). Thus, data are available in different resolutions and methods needs to be devised to work with multiple resolutions of the data. However, current standard algorithms only work with a single resolution of data. So, sampling in different resolutions possesses

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 86–97, 2010.

high importance. In this paper, we model multiresolution data and use statistical significance testing on data generated by generative models. Finite mixture models are generative models [2,3] able to generate the potentially observable data. Over the years, finite mixture models have been extensively used in many application domains including model based clustering, classification, image analysis, and collaborative filtering in analysis of high dimensional data because of their versatility and flexibility. In spite of the wide application areas of mixture models, the evaluation of mixture models are often based on the likelihood of the model on the original data, not by testing the data generated by the generative models.

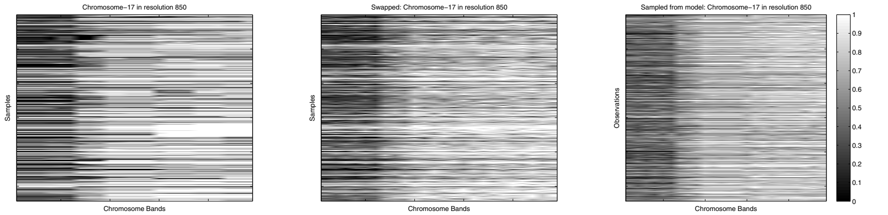
In [4], the authors used HMO (Hypothetical Mean Organism) motivated from Bacteriology [5] and maximal frequent itemsets[6] to define the data to the domain experts in a compact and understandable manner. Furthermore, in [7], the authors also compared the frequent itemsets [8,9] extracted from each cluster to that extracted globally showing that the frequent itemsets were significantly different. However, the authors failed to consider the significance of the itemsets and their preservation by generative models. Study of patterns generated by the generated models has received little interest. However, preserving patterns from the original data should be essentially an important property of mixture models and if properly designed can be one of the benchmarks for selecting better mixture models. In this paper, we experiment with finite mixture models of multivariate Bernoulli distribution to test whether the statistically significant itemsets are preserved by mixture models. We also extend the ideas in [10] to observe if the significant itemsets are preserved by the sampling in different resolutions.

Novelties in this paper are determination of presence of statistically significant itemsets with respect to sampling different resolutions and especially by the data generated through the generative mixture models. Furthermore, we experiment the mixture model with different levels of noise showing that the trained mixture models are robust to noise in lower resolution and when there is significant amount of data to train and constrain the mixture model thus showing the importance of working in multiple resolutions which is useful for database integration.

Rest of the paper is organized as follows: Section 2 presents the dataset used in the experiments. Section 3 reviews the theoretical framework for experiments including sampling, randomization and mixture modelling. Section 4 explicates the experiments performed on the data and discusses the obtained results. Section 5 draws conclusions from the experimental results.

2 DNA Copy Number Amplification Dataset

The dataset used in the experiments defines DNA amplifications in different chromosomes. Amplification is the special case of duplication where the copy number increases more than 5 [11]. The data was collected by bibliomics survey of 838 journal articles during 1992-2002 by hand without using state-of-the-art



(a) Original Data (b) Randomized Data (c) Sampled from Model

Fig. 1. DNA copy number amplifications in chromosome-17, resolution 850. $\overline{\underline{\boldsymbol{X}}} = (X_{ij})$, $X_{ij} \in \{0, 1\}$. Each row represents one sample of the amplification pattern for a patient and each column represents one of the chromosome bands.

text mining techniques [4,12]. The dataset contained information about the amplification patterns of 4590 cancer patients in resolution 400. There was another set of similar data but in resolution 850 with higher sample size. The dataset shown in Figure 1 contains the original data in resolution 850, the randomized version and sampled from the mixture model. Each row describes one sample of cancer patient while each column identifies one chromosome band(region). The amplified chromosome regions were marked with 1 while the value 0 defines that the chromosome band is not amplified. Patients whose chromosomal band had not shown any amplification for specific chromosome were not included in the experiments since we are interested in modelling the amplifications, not their absence.

3 Theoretical Framework

Determining the significance of the results obtained by any algorithm or method is an actively researched area. Statistical significance testing have often been implemented to determine the significance of the results. In this paper, we implement our statistical significance testing on data in multiple resolutions and data generated by mixture models.

3.1 Sampling Resolutions

We have recently in [10] suggested three downsampling and a simple upsampling technique for 0-1 data and performed experiments on them showing that the methods are fairly similar. Upsampling is the process of changing the resolution of data from coarse resolution to finer resolution and downsampling is the process of changing the resolution of data from fine resolution to coarse resolution. Upsampling makes multiple copies of similar chromosome bands in higher resolution. Downsampling, in turn, proceeds with one of three different methods: OR-function, Majority decision and Weighted Downsampling. In OR-function downsampling, a cytogenetic band in lower resolution is amplified if any of the bands in higher resolution which combines to form the cytogenetic band in the lower resolution is amplified. In majority decision downsampling method, the cytogenetic band in lower resolution is amplified if majority of the cytogenetic band in higher resolution are amplified. In weighted downsampling method, the length of the cytogenetic bands are considered. The cytogenetic band in lower resolution is amplified if the total length of amplified band is higher than that of the unamplified band.

3.2 Randomization

Statistical significance testing on datasets are not trivial as the data belongs to a class of empirical distributions thus integrating over the PDF(Probability Density Function) to calculate the $p-$values is often not possible. Furthermore, given the data set $\mathcal{D}$, its PDF or true generating model is often unknown. It is trivial to integrate over the empirical distribution where a null distribution can be fixed and samples can be drawn from the null distribution. Randomization [13] is one of the method to sample from null distribution and it has been proposed with some plausible results and implemented in various application areas such as redescription mining [14]. Comparing segmentations of genomic sequences [15] among many others.

Consider a 0-1 dataset, $\mathcal{D}$ with m rows and n columns. Let $\mathcal{D}_1, \mathcal{D}_2 \ldots \mathcal{D}_n$ be the randomized data produced using the randomization approach repeated n times. Also, consider a data mining algorithm $\mathcal{A}$, for instance frequent set mining and mixture modelling in our case which is run on the data $\mathcal{D}$ with the result $\mathcal{A}(\mathcal{D})$. The result $\mathcal{A}(\mathcal{D})$ determines the structural measure of the dataset $\mathcal{D}$, the frequencies of frequent itemset and likelihood in our case. The randomized datasets $\mathcal{D}_1, \mathcal{D}_2 \ldots \mathcal{D}_n$ are also subjected to the algorithm $\mathcal{A}$ producing results $\mathcal{A}(\mathcal{D}_1), \mathcal{A}(\mathcal{D}_2) \ldots \mathcal{A}(\mathcal{D}_n)$. The task is then to determine whether the result on the original data is different from the results on the randomized data. Empirical $p-$values can be used for the same purpose.

Null Distribution: Given a binary dataset $\mathcal{D}$, the null distribution considered in the paper are all the datasets satisfying all the following properties:

1. The dataset of the same size i.e. number of rows and columns of randomized data is equal to the number of rows and columns of the original data.
2. The dataset with same row and column margins. Margins here describes the sums. Thus, row and column sums are exactly fixed. This automatically preserves the number of ones in the dataset i.e. the number of amplifications.

As the the constraints discussed above increases, the randomization is becomes more conservative. However, the main focus is to compare the results obtained with the original dataset with closely related datasets. Furthermore, the number of datasets satisfying the above constraints are still significantly high. Generally, the application area determines the constraints of the randomization. Maintaining row and column margins in this case is adapted from the idea in [13] which seems relevant in our case considering the fact that most of the binary datasets especially in the field of biology such as the amplification data discussed in Section 2 are often spatially dependent and sparse. On the other hand, if the randomization is not subjected to the constraints discussed above then any result of an algorithm turns out to be relevant. With lesser constraints, the number of randomized datasets to sample for convergence discussed in Section 4.1 increases which consequently increases the computational complexity of the approach. Experimental results in [13] have shown that complexity of using a data mining algorithm $\mathcal{A}$ on a dataset has significantly higher computational complexity compared to the generation of randomized dataset under the constraints discussed above. Similar to [13], the data is randomomized in the with repeated 0-1 swaps until convergence. The null hypothesis H_0 throughout this paper is that for all datasets $\mathcal{D}$ that satisfies the given constraints, the test statistic follows the same distribution. Test statistic used here is frequency or the support (α) in case of frequent itemset and sample likelihood in case of mixture models.

***p*−Values:** p-value can be defined as probability of obtaining a test statistic at least as extreme as the one that was actually observed, assuming that the null hypothesis is true [16,17]. Let $\hat{\mathcal{D}} = \{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_k\}$ be the randomized versions, sampled i.i.d from the null distribution, of the original data $\mathcal{D}$. The one-tailed *empirical p−value* of $\mathcal{A}(\mathcal{D})$ for $\mathcal{A}(\mathcal{D})$ being large is

$$\tilde{p} = \frac{1}{n+1} \left(\sum_{i=1}^{n} I(\mathcal{A}(\mathcal{D}_i) \geq \mathcal{A}(\mathcal{D})) + 1 \right), \tag{1}$$

where $i \in \{1, 2 \ldots k\}$ and I is the indicator variable.

The Equation 1 gives the fraction of randomized dataset whose structural measure, itemset frequency (support) in case of frequent itemset and sample likelihood in case of mixture models, is greater than the original data $\mathcal{A}(\mathcal{D})$. In one-tailed $p-$value small value of $\mathcal{A}(\mathcal{D})$ are interesting and can be defined similarly for the two-tailed test. In this paper the randomized datasets are produced using Markov Chain Monte Carlo(MCMC) approach. The samples produced by MCMC are not independent thus diminishing the reliability of the $p-$values. To mitigate this problem and guarantee the ex-changeability of samples, we implement forward-backward approach discussed in [18]. The basic idea is to run the chain, a number of defined steps, say J backwards and forward after reaching J. In other words, given the original dataset $\mathcal{D}$, a dataset $\hat{\mathcal{D}}$ is obtained such that the path length between $\mathcal{D}$ and $\hat{\mathcal{D}}$ is J. The desired number of $\mathcal{K}$ samples of randomized data is obtained by running the chain J steps forward and obtaining the samples $\hat{\mathcal{D}}_i$ thus producing $\mathcal{D}, \hat{\mathcal{D}}_1 \ldots \hat{\mathcal{D}}_k$ as the set of exchangeable samples. Furthermore, the $p-$values were adjusted for multiple hypothesis testing using the Holm-Bonferroni test correction[19].

3.3 Mixture Models of Multivariate Bernoulli Distribution

Cancer is not a single disease but a collection of several diseases. Furthermore, the amplification data discussed in Section 2 being high dimensional binary data, finite mixtures of multivariate Bernoulli distribution was selected as the model to model the amplification data. The finite mixture of multivariate Bernoulli distributions is defined as:

$$p(\mathcal{D}|\boldsymbol{\Theta}) = \sum_{j=1}^{\mathrm{J}} \pi_j \prod_{i=1}^{\mathrm{d}} \theta_{ji}^{x_i} (1 - \theta_{ji})^{1-x_i}, \tag{2}$$

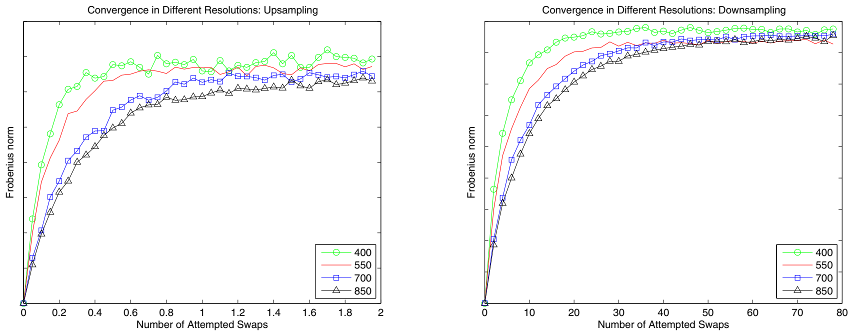
where the data is assumed to originate from the known number of components J. The mixture proportions π_j satisfy the properties such as convex combination such that $\pi_j \geq 0$ and $\sum_{j=1}^{\mathrm{J}} \pi_j = 1, \forall j = 1, \ldots J$. The model parameters $\boldsymbol{\Theta}$ is composed of $\theta_1, \theta_2, \theta_3 \ldots \theta_d$ for each component distribution.

Model selection in finite mixture modelling refers to the process of selecting number of mixture components, J in the data. 10-fold cross-validation [20,21] is used to select the optimal number of components taking parsimony into account. The process of model selection employed is similar to [4,10,22]. Since the mixture models are complex and sample size of data was small to constrain it, chromosome-wise mixture modelling was performed for data in different resolutions. Expectation Maximization algorithm [23,24] was used to train the mixture models using BernoulliMix[25] which is an open source program package for finite mixture modelling of multivariate Bernoulli distribution.

4 Experiments

4.1 Convergence Analysis of the Swaps

In order to determine the optimal number of swaps to be performed, convergence test for the randomized data was performed. In our experiments, the process of randomization is said to converge when the distance between the the original data and the randomized data changes the least with respect to the predefined difference measure. Similar to [13] and [26], the distance measure used here is the Frobenius norm between the original and the randomized matrix. In order to test the convergence, first the number of attempted swaps is fixed to 1 and increased by the step size of 1. The approach used here differs from [13] and [26] because they set the initialization point to $\mathcal{K}$ equal to the number of ones in the data and increase the number of attempts in multiples of $\mathcal{K}$. Such approach could prove beneficial in large datasets but since amplification dataset is small, it was very easy to compute the swaps thus making it easier to initialize number of attempted swaps to 1. Furthermore, similar dataset was available in resolution 850 with higher sample size. Thus, the convergence test was performed for both the data and their upsampled and downsampled versions as shown in Figure 2. Ten different instances of the swaps are performed and the mean of the results is taken as the final convergence test. Similar, convergence analysis was also performed for combined data and the sampled data. Convergence of sampled data was similar to the original data from which the model was trained. However, in case of combined data, convergence required relatively higher number of swaps i.e. 700000 swaps. Figure 2 shows that the swap converges when the number of



(a) Resolution 400 and Upsampled

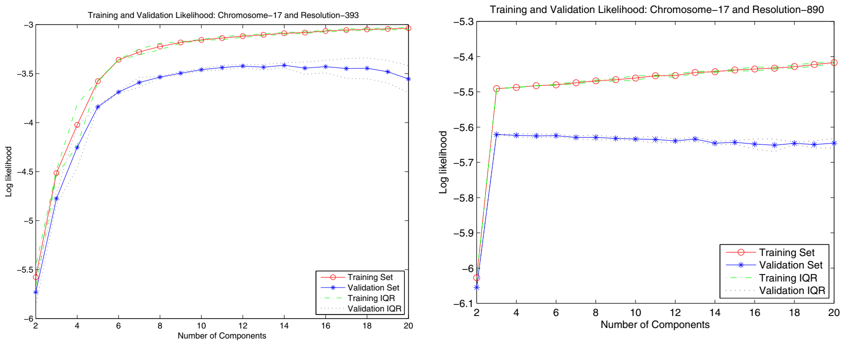
(b) Resolution 850 and Downsampled

Fig. 2. Convergence analysis for randomization with respect to 0-1 swaps

attempted swaps is approximately 16000 for original data in resolution 400. From the Figure 2 it can also be seen that the Frobenius norm increases rapidly until certain number of attempted swaps and then tends to stabilize. The stabilizing point is taken as the convergence. As discussed in Section 2, the sample size of data in resolution 850 was high thus taking longer time to converge. The number of swap attempted to get the randomized data in this case is 600000.

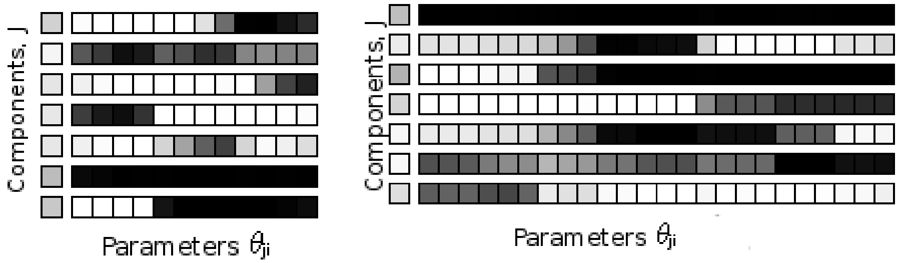
4.2 Model Selection in Mixture Model

Model selection in the context of mixture modelling is the selection of number of components of the mixture model. It is often recommended to repeat cross-validation technique a number of times, at least 10, because a 10-fold cross-validation can be seen as a "standard" measure of the performance whereas ten 10-fold cross-validations would be a "precise" measure of performance[27]. In addition, EM-algorithm is highly sensitive to initializations and the global optimum is not often guaranteed [28]. Therefore, the cross-validation procedure was repeated 50 times. Since the analysis was performed chromosome-wise, the data dimension was relatively less. Thus, the number of mixture components were varied between 2 and 20. Using higher number components can overfit the data. Furthermore, our major goal, as in [4], was to generate compact and parsimonious models. The log-likelihood was averaged for each component and the interquartile range(IQR) was calculated. Furthermore, the model selection procedure was also performed for the randomized data. In Figure 3a, both training and validation likelihood are smoothly increasing curves with low variation in IQR. The number of components selected in this case is 7, taking the parsimony into account. We also performed similar model selection procedure on the randomized data as shown in Figure 3b. It was found that there is no well defined clustering structure present in the data with respect to the mixture models. Furthermore, the results on randomized data also proves that the data is not a random data but there is a well-defined structure present in the data which mixture model is able to extract.



(a) Original combined Resolution 400 (b) Randomized combined Resolution 400

Fig. 3. Model Selection procedure and Model visualization: Example case in combined data of Chromosome-17 in resolution-400 and its corresponding randomized version. Corresponding IQR (Inter Quartile Range) for each training and validation run has also been plotted.



(a) Resolution 400

(b) Resolution 850

Fig. 4. Two different models for the combined data trained in resolution 400 and 850

After selecting the number of components, ten different models were trained to convergence and best of the trained models were used to calculate the likelihood on data as shown Figure 5b. The model was also used to sample the data to calculate the significant itemsets in the sampled data. Figures 4a and 4b are the final models trained to convergence for combined data in resolution 400 and 850 respectively. Similarity of the models can be tracked visually from the model visualization as in Figure 4. For example, component 6 in Figure 4a corresponds to component 1 in 4b.

4.3 Significance of Frequent Itemsets and Data Samples

In the experimental setup, first the frequencies of the itemsets of the size two were determined from the original data. The itemsets of size three and above were discarded from the experiments for simplicity and space constraints for explaining the results. However, results in [10] has shown that generally the frequent itemsets in the amplification data discussed in Section 2 are large and consecutive. The core of the work was to determine if the statistically significant itemsets were preserved in different resolutions and by the generative mixture model. First the itemsets of size two which had a frequency or support($\alpha \approx 0.5$) were determined and the original data was then subjected to randomization. Randomization produces 100000 samples of randomized dataset. Larger number of random samples are chosen because Holm-Bonferroni [19] used to correct for multiple hypothesis requires higher number of samples for plausible results. The structural measure used to calculate the $p-$values in our case is the support or the frequency of the itemsets. The choice of frequency or support($\alpha \approx 0.5$) is arbitrary but motivated by majority voting protocol and constraining the number of frequent itemsets thus making it easier to interpret and report. Furthermore, itemsets with very low support but statistically significant are not highly interesting. The samples of data were generated equal to the number of samples in the original data. Similarly, the data generated from the trained mixture models were also subjected to randomization to determine the statistically significant itemsets.

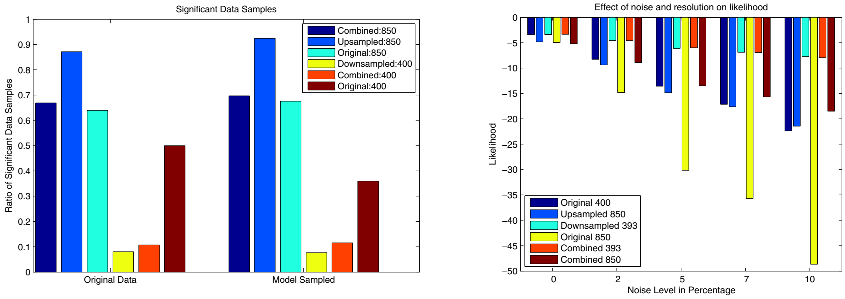
Table 1. Itemsets of size 2 with their frequency (support) in original as well as sampled resolution. Results of Downsampling have been omitted because of space constraints. The symbol ${}^{item}_{n}\boldsymbol{C}^{item}_{r}$ suggests combination where subscript n and r determines n choose r in the combination and superscript determines the item to start and end the combination.

Significant itemsets of Size 2 at $\alpha = 0.05$			
Data	**Support**	**Original Data**	**Model Sampled**
Original 393	.4	{9,10}, {11,12}	{9,10}, {11,12}
Upsampled 850	.4	${}^{10}_{5}\boldsymbol{C}^{14}_{2}$, ${}^{15}_{4}\boldsymbol{C}^{18}_{2}$, ${}^{19}_{6}\boldsymbol{C}^{24}_{2}$	${}^{10}_{5}\boldsymbol{C}^{14}_{2}$, ${}^{19}_{6}\boldsymbol{C}^{24}_{2}$
Combined 393	.6	{ 5, 7}, { 5, 12}, ${}^{8}_{6}\boldsymbol{C}^{12}_{2}$	{ 5, 7}, { 5, 12}, { 7, 12}, ${}^{8}_{6}\boldsymbol{C}^{12}_{2}$
Combined 850	.6	${}^{10}_{6}\boldsymbol{C}^{15}_{2}$, {12,16}, {12,17}, {12,18}, {13,16}, {13,17}, {13,18}, {14,16}, {14,17}, {14,18}, ${}^{15}_{10}\boldsymbol{C}^{24}_{2}$	${}^{10}_{6}\boldsymbol{C}^{15}_{2}$, {12,16}, {12,17}, {12,18}, {12,20}, {13,16}, {13,17}, {13,18}, {13,20}, {14,16}, {14,17}, {14,18}, {14,20}, ${}^{15}_{10}\boldsymbol{C}^{24}_{2}$

The $p-$values were calculated to test the significance of the itemsets. The statistically significant itemsets computed at significance level (α)= 0.05 in the original data and the sampled data from the model is compared and analyzed. Table 1 shows that significant itemsets are approximately but not exactly preserved by the generative mixture model as well as the sampling of resolutions. Difference is subtle in higher resolution. The itemsets in lower resolution correspond to itemsets in higher resolution. For example, itemset {11,12} in resolution 400 corresponds to itemset ${}^{19}_{6}\boldsymbol{C}^{24}_{2}$ in resolution 850. It is to be noted that not all frequent itemsets are significant and not all significant itemset are frequent. For example, in case of combined resolution 400, itemset {1,2} is significant where as it is not frequent. Furthermore, itemset {7,12} is frequent but not significant.

We also determined the number of significant data samples in different resolutions and from the sampled model. Figure 5a suggests that numbers of significant data vectors are preserved in the generative models. During our experiments, we also determined the indices of the significant data vectors and it was seen that indices of the significant vectors are not preserved i.e. generated of samples of data are not arranged in similar manner to original data. Furthermore, it was also seen that finer resolution has higher number of significant data samples because with increasing dimension the uniqueness of the rows increases and the 0-1 swap strategy used in the randomization ceases to function properly. However, this has little or no significance because of i.i.d assumption for each data sample.

4.4 Effect of Noise on the Likelihood



(a) Significant Data Samples

(b) Effect of Noise

Fig. 5. Ratio of significant data samples to the number of samples in the left panel and effect of noise and resolution on the likelihood in right panel

We added random noise to the data. Since the data was binary data, adding noise is simply flipping the bits i.e. changing ones to zeros and zeros to ones. Addition of 5% noise means that 5% of total data items in the dataset are flipped. Figure 5b shows that the effect of noise will be significantly higher for data in finer resolution than the data in the lower resolution. Furthermore, when the number of samples is low (Cases: Original 400 and Upsampled to 850), the difference in the likelihood is large because the number of samples are too low to constrain the mixture model. However, when the number of samples are increased, as in case of combined datasets, the variation in likelihood is not significant. Nevertheless, likelihood for the data in the higher resolution deviates significantly even when the sample size is increased.

5 Summary and Conclusions

We use statistical significance testing on data in different resolutions and on data generated by the generative mixture models using randomization. From the experiments we conclude that finite mixtures of multivariate Bernoulli distribution retains the significant itemsets and the significant data vectors in the original data even when the mixture model is trained parsimoniously. Furthermore, experiments with different levels of noise on the data shows that models parsimonious models in coarse resolution are more robust to noise. Nevertheless, when there is adequate amount of data to constrain the mixture model, the effect of noise diminishes significantly even in higher resolution.

References

1. Shaffer, L.G., Tommerup, N.: ISCN 2005: An International System for Human Cytogenetic Nomenclature(2005) Recommendations of the International Standing Committee on Human Cytogenetic Nomenclature. Karger (2005)
2. McLachlan, G.J., Peel, D.: Finite mixture models. In: Probability and Statistics – Applied Probability and Statistics Section, vol. 299. Wiley, New York (2000)
3. Everitt, B.S., Hand, D.J.: Finite mixture distributions. Chapman and Hall, Boca Raton (1981)
4. Hollmén, J., Tikka, J.: Compact and understandable descriptions of mixtures of bernoulli distributions. In: R. Berthold, M., Shawe-Taylor, J., Lavrač, N. (eds.) IDA 2007. LNCS (LNAI), vol. 4723, pp. 1–12. Springer, Heidelberg (2007)
5. Gyllenberg, M., Koski, T.: Probabilistic models for bacterial taxonomy. International Statistical Review 69, 249–276 (2000)
6. Burdick, D., Calimlim, M., Gehrke, J.: Mafia: A maximal frequent itemset algorithm for transactional databases. In: ICDE, pp. 443–452 (2001)
7. Hollmén, J., Seppänen, J.K., Mannila, H.: Mixture models and frequent sets: Combining global and local methods fordata. In: SDM (2003)
8. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data, pp. 207–216. ACM, New York (1993)
9. Mannila, H., Toivonen, H., Verkamo, A.I.: Efficient algorithms for discovering association rules. In: Fayyad, U.M., Uthurusamy, R. (eds.) AAAI Workshop on Knowledge Discovery in Databases (KDD-94), Seattle, Washington, pp. 181–192. AAAI Press, Menlo Park (1994)
10. Adhikari, P.R., Hollmén, J.: Patterns from multiresolution 0-1 data. In: UP '10: Proceedings of the 16th ACM SIGKDD. ACM, New York (to appear, 2010)
11. Bishop, J.F.: Cancer facts: a concise oncology text. Harwood Academic Publishers, Amsterdam (1999)
12. Myllykangas, S., Tikka, J., Böhling, T., Knuutila, S., Hollmén, J.: Classification of human cancers based on DNA copy number amplification modeling. BMC Medical Genomics 1, 15 (2008)
13. Gionis, A., Mannila, H., Mielikäinen, T., Tsaparas, P.: Assessing data mining results via swap randomization. ACM Transactions on Knowledge Discovery from Data 1(3), 14 (2007)
14. Gallo, A., Miettinen, P., Mannila, H.: Finding subgroups having several descriptions: Algorithms for redescription mining. In: SDM, pp. 334–345 (2008)
15. Haiminen, N., Mannila, H., Terzi, E.: Comparing segmentations by applying randomization techniques. BMC Bioinformatics 8(1), 171 (2007)

16. Schervish, M.J.: P values: What they are and what they are not. American Statistician 50(3), 203–206 (1996)
17. De La Horra, J., Rodriguez-Bernal, M.T.: Posterior predictive p-values: What they are and what they are not. Test 10(1), 75–86 (2001)
18. Besag, J., Clifford, P.: Generalized monte carlo significance tests. Biometrika 76(4), 633–642 (1989)
19. Holm, S.: A simple sequentially rejective multiple test procedure. Scandinavian Journal of Statistics 6, 65–70 (1979)
20. Geisser, S.: A predictive approach to the random effect model. Biometrika 61(1), 101–107 (1974)
21. Monsteller, F., Tukey, J.: Data analysis including statistics. In: Lindzey, G., Aronson, E. (eds.) Handbook of Social Psychology, vol. 2. Addison-Wesley, Reading (1968)
22. Tikka, J., Hollmén, J., Myllykangas, S.: Mixture modeling of DNA copy number amplification patterns in cancer. In: Sandoval, F., Prieto, A.G., Cabestany, J., Graña, M. (eds.) IWANN 2007. LNCS, vol. 4507, pp. 972–979. Springer, Heidelberg (2007)
23. Wolfe, J.H.: Pattern clustering by multivariate mixture analysis. Multivariate Behavioral Research 5, 329–350 (1970)
24. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, Series B 39(1), 1–38 (1977)
25. Hollmén, J.: BernoulliMix: Program package for finite mixture models of multivariate Bernoulli distributions (May 2009), http://www.cis.hut.fi/jHollmen/BernoulliMix/
26. Hanhijärvi, S., Ojala, M., Vuokko, N., Puolamäki, K., Tatti, N., Mannila, H.: Tell me something I don't know: randomization strategies for iterative data mining. In: KDD '09: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 379–388. ACM, New York (2009)
27. Gay, S.D.: Datamining in proteomics: extracting knowledge from peptide mass fingerprinting spectra. PhD thesis, University of Geneva, Geneva (2002)
28. Mclachlan, G.J., Krishnan, T.: The EM Algorithm and Extensions, 1st edn. Wiley Interscience, Hoboken (November 1996)

Novel Machine Learning Methods for MHC Class I Binding Prediction

Christian Widmer[1,*], Nora C. Toussaint[2,*], Yasemin Altun[3],
Oliver Kohlbacher[2], and Gunnar Rätsch[1]

[1] Friedrich Miescher Laboratory, Max Planck Society, Spemannstr. 39, 72076
Tübingen, Germany
[2] Center for Bioinformatics Tübingen, Eberhard-Karls-Universität, Sand 14, 72076
Tübingen, Germany
[3] Max Planck Institute for Biological Cybernetics, Spemannstr. 38, 72076
Tübingen, Germany

Abstract. MHC class I molecules are key players in the human immune system. They bind small peptides derived from intracellular proteins and present them on the cell surface for surveillance by the immune system. Prediction of such MHC class I binding peptides is a vital step in the design of peptide-based vaccines and therefore one of the major problems in computational immunology. Thousands of different types of MHC class I molecules exist, each displaying a distinct binding specificity. The lack of sufficient training data for the majority of these molecules hinders the application of Machine Learning to this problem.

We propose two approaches to improve the predictive power of kernel-based Machine Learning methods for MHC class I binding prediction: First, a modification of the Weighted Degree string kernel that allows for the incorporation of amino acid properties. Second, we propose an enhanced Multitask kernel and an optimization procedure to fine-tune the kernel parameters. The combination of both approaches yields improved performance, which we demonstrate on the IEDB benchmark data set.

1 Introduction

Despite the success of traditional whole-organism vaccines in the last century there is still a lack of effective vaccines for many diseases, for example AIDS and cancer. A fairly new approach to vaccination, the peptide-based vaccines, shows great promise here. Peptide-based vaccines utilize peptides, i.e. small protein fragments, derived from, e.g., viral proteins to induce immunity. In order for a peptide to trigger an immune response from inside a host's cell, it has to bind to a major histocompatibility complex class I (MHC-I) molecule. The resulting peptide/MHC-I complex will be transported to the cell surface where it can be recognized by specific immune system cells, the T cells (Fig. 1A), and thereby induce an immune response. Thus, MHC-I binding is a prerequisite for peptide immunogenicity. Furthermore, identifying peptides with a high affinity to

[*] Authors contributed equally.

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 98–109, 2010.

MHC-I molecules is generally considered the best way to identify immunogenic peptides. Since only immunogenic peptides are suitable candidates for inclusion in a peptide-based vaccine, the prediction of peptides binding to MHC-I is of great interest in the field of vaccine design.

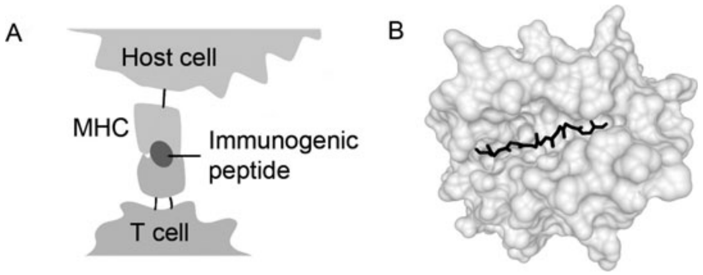


Fig. 1. Peptide/MHC-I complex. A) An MHC-I molecule presents an immunogenic peptide on the cell surface where it is recognized by a T cell. B) The structure of a nonameric peptide complexed with an MHC-I molecule. The binding groove is closed at both ends and the peptide is bound in an extended conformation. (PDB-ID: 3L3D (http://www.pdb.org) [2], plotted with BALLView [11])

MHC-I molecules are membrane-bound proteins with a closed binding groove that holds peptides in an extended conformation (Fig. 1B). They typically bind peptides that contain eight to ten amino acids (AAs) with a preference for nine AAs. The corresponding gene complex is highly polymorphic. As of today, more than 3,000 different MHC-I alleles are known,each coding for an MHC-I molecule binding a specific range of peptides. Any human has up to six different types of MHC-I molecules. This implies that a peptide that is capable of inducing an immune response in one individual might never be presented on the cell surface in another. In order to design vaccines effective for a given population, it is therefore necessary to accurately predict MHC-I binding for a wide range of different MHC alleles [22].

Many computational methods for the classification of peptides into MHC-I binders or non-binders have been proposed: ranging from matrices [14, 17] to machine learning [3, 1].All of these methods require a certain amount of experimental binding data for each allele under consideration. However, a major problem in MHC-I binding prediction is the lack of data: for the vast majority of the known alleles there is no or only little experimental binding data available yielding the development of accurate prediction methods for most alleles rather challenging. In 2008, Laurent and Vert [8] proposed a kernel-based approach that tries to overcome the lack of training data by sharing binding information across alleles.

In this work, we propose two approaches to improve MHC-I binding prediction. First, we consider an improved string kernel, which takes AA properties into account and thereby allows more accurate predictions for alleles with little binding data. Second, we consider an enhanced Multitask learning algorithm, which can be used to improve prediction performance for an allele by utilizing binding data of similar alleles. We are able to show that the combination of both approaches outperforms existing methods.

2 Improved String Kernels for MHC-I Binding Prediction

Background. String kernels are a very powerful tool for machine learning in bioinformatics due to their capability to exploit the sequential structure of AA or nucleotide sequences. They have been successfully applied to various problems in computational biology, ranging from protein remote homology detection [10],to gene identification [16,20], to drug design [8].

MHC-I molecules bind peptides in an extended conformation (Fig. 1B). Within the complex the peptide's side chains interact with surrounding side chains of the MHC and also with each other. Each of the peptide's side chains contributes to the binding affinity. The respective contribution is influenced by the position of a side chain within the peptide sequence as well as by the AA types of its neighboring side chains. A string kernel is very well suited to handle such data is the *Weighted Degree (WD) kernel* [15]. The WD kernel considers sequences of fixed length L and counts co-occurring substrings in both sequences at the same position. It is defined as

$$K_\ell^{\mathrm{wd}}(\boldsymbol{x}, \boldsymbol{z}) = \sum_{d=1}^{\ell} \beta_d \sum_{i=1}^{L-d+1} \mathbf{I}\left(\boldsymbol{x}_{[i:i+d]} = \boldsymbol{z}_{[i:i+d]}\right) \qquad (1)$$

where $\beta_d = 2\frac{\ell-d+1}{\ell^2+\ell}$ is the weighting of the substring lengths.

A major downside to using string kernels on AA sequences is that prior knowledge on properties of individual AAs, e.g., their size, hydrophobicity, charge, cannot be easily incorporated. Especially when dealing with small training data sets as common in MHC-I-binding prediction, inclusion of this information in the sequence representation would be beneficial.

A straightforward approach to utilizing this knowledge is to consider a representation of the sequence as vector of the physico-chemical properties of all sequence elements, i.e. AAs. One may then use a standard kernel to compute sequence similarities, as, e.g., done in [24,13]. This approach, however, ignores the sequential nature of the underlying data.

Here, we propose to combine the benefits of standard string kernels with the ones of physico-chemical descriptors for AAs.

Idea. As string kernels in general, the WD kernel (1) compares substrings of length ℓ between the input sequences $\boldsymbol{x}$ and $\boldsymbol{z}$. We can rewrite the corresponding term $\mathbf{I}(\overline{\boldsymbol{x}} = \overline{\boldsymbol{z}})$ as:

$$\mathbf{I}(\overline{\boldsymbol{x}} = \overline{\boldsymbol{z}}) = \left\langle \Phi_\ell(\overline{\boldsymbol{x}}), \Phi_\ell(\overline{\boldsymbol{z}}) \right\rangle ,$$

where $\overline{\boldsymbol{x}}, \overline{\boldsymbol{z}} \in \Sigma^\ell$ and $\Phi_\ell : \Sigma^\ell \rightarrow \mathbb{R}^{|\Sigma^\ell|}$.

$\varPhi_\ell(\overline{\boldsymbol{x}})$ can be indexed by a substring $s \in \varSigma^\ell$ and is defined as $\varPhi_\ell(\overline{\boldsymbol{x}})_s = 1$, if $s = \overline{\boldsymbol{x}}$, and 0 otherwise. Using $\varPhi_1 : \varSigma \mapsto \{0,1\}$, a simple encoding of the letters into $|\varSigma|$-dimensional unit vectors, the substring comparison can be rewritten as

$$\mathbf{I}(\overline{\boldsymbol{x}} = \overline{\boldsymbol{z}}) = \prod_{l=1}^{\ell} \langle \varPhi_1(\overline{\boldsymbol{x}}_l), \varPhi_1(\overline{\boldsymbol{z}}_l) \rangle \,,$$

$\varPhi_1$ ignores the relations between the letters in the alphabet. Since this is a problem when considering AAs, we replace $\varPhi_1$ with a feature map $\varPsi$ that takes relations between the AAs into account. This leads to the following kernel on AA substrings:

$$K_\ell^\Psi(\overline{\boldsymbol{x}}, \overline{\boldsymbol{z}}) = \prod_{l=1}^{\ell} \langle \varPsi(\overline{\boldsymbol{x}}_l), \varPsi(\overline{\boldsymbol{z}}_l) \rangle \,. \tag{2}$$

Using the feature representation corresponding to this kernel, we can now recognize sequences of AAs that have certain properties (e.g. first AA: hydrophobic, second AA: large, third AA: positively charged, etc.): For every combination of products of features involving exactly one AA property per substring position, there is one feature induced in the kernel. A richer feature space including combinations of several properties from every position can be obtained using the following two formulations. The first is based on the polynomial kernel:

$$K_{\ell,d}^\Psi(\overline{\boldsymbol{x}}, \overline{\boldsymbol{z}}) = \left(\sum_{l=1}^{\ell} \langle \varPsi(\overline{\boldsymbol{x}}_l), \varPsi(\overline{\boldsymbol{z}}_l) \rangle \right)^d \,, \tag{3}$$

and the second on the RBF kernel:

$$K_{\ell,\sigma}^\Psi(\overline{\boldsymbol{x}}, \overline{\boldsymbol{z}}) = \exp\left(-\frac{\sum_{l=1}^{\ell} \|\varPsi(\overline{\boldsymbol{x}}_l) - \varPsi(\overline{\boldsymbol{z}}_l)\|^2}{\sigma^2} \right). \tag{4}$$

Improved WD Kernel. Replacing the substring comparison $\mathbf{I}(\overline{\boldsymbol{x}} = \overline{\boldsymbol{z}})$ in (1) with one of the formulations in (2), (3), or (4) together with a set of features $\varPsi(a)$ for each letter $a \in \varSigma$ (i.e. for each AA), directly leads to a generalized form of the WD kernel:

$$K_\ell^{\mathrm{wd},\Psi}(\boldsymbol{x}, \boldsymbol{z}) = \sum_{d=1}^{\ell} \beta_d \sum_{i=1}^{L-d+1} K_d^\varPsi(\boldsymbol{x}_{[i:i+d]}, \boldsymbol{z}_{[i:i+d]}). \tag{5}$$

$K_\ell^{\mathrm{wd},\Psi}$ is a linear combination of kernels and therefore a valid kernel [18]. It can be computed efficiently, with a complexity comparable to that of the original WD kernel.

The combination of the WD kernel with the RBF substring kernel (4) is particularly interesting:

$$K_{\ell,\sigma}^{\mathrm{wd},\Psi}(\boldsymbol{x}, \boldsymbol{z}) = \sum_{d=1}^{\ell} \beta_d \sum_{i=1}^{L-d+1} \exp\left(-\frac{\sum_{j=1}^{d} \|\varPsi(\boldsymbol{x}_j) - \varPsi(\boldsymbol{z}_j)\|^2}{\sigma^2} \right). \tag{6}$$

For a bijective encoding Ψ and $\sigma \rightarrow 0$, this *WD-RBF kernel* corresponds to the WD kernel: the RBF substring kernel will be one only for identical substrings, otherwise it will be zero. Thus, employing the WD-RBF kernel will, at least in theory, always yield equal or better performances than the original WD kernel.

3 A New Multitask Kernel for MHC-I Binding Prediction

We will build upon a kernel-based formulation of Multitask Learning, as proposed by [4]:

$$\begin{aligned} &\max_{\alpha} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \tilde{K}((\mathbf{x}_i, s), (\mathbf{x}_j, t)) + \sum_{i=1}^{n}\alpha_i \qquad (7)\\ &\text{s.t.} \quad \alpha^T\mathbf{y} = 0, \quad 0 \leq \alpha_i \leq C \ \forall i \in \{1, n\}, \end{aligned}$$

where s and t correspond to the tasks associated with examples x_i and x_j, respectively.

$$\tilde{K}((\boldsymbol{x}_i, s), (\boldsymbol{x}_j, t)) = K(\boldsymbol{x}_i, \boldsymbol{x}_j) + K^{\text{dirac}}(s, t) \cdot K(\boldsymbol{x}_i, \boldsymbol{x}_j), \qquad (8)$$

where K denotes the base kernel that captures the interactions between examples from all tasks and $K^{\text{dirac}}(s, t)$ is defined as

$$K^{\text{dirac}}(s, t) = \begin{cases} 1, \text{ if } t = s \\ 0, \text{ else} \end{cases}. \qquad (9)$$

It was shown in previous work [7] that it pays off to use multitask learning methods for the problem of MHC-I binding prediction. In particular, a multitask kernel based on the product of allele (i.e. task) similarity and peptide (i.e. instance) similarity was used:

$$K^{\text{MT}}((x, s), (z, t)) = K^{\text{all}}(s, t) \cdot K^{\text{pep}}(x, z),$$

which is a generalization of the kernel presented in Equation (8). Here, the similarity between tasks is explicitly taken into account, instead of solely setting a higher default similarity for in-domain comparisons. In the case of MHC-I molecules, the pseudo sequence (i.e. the AAs in the binding groove of the MHC that interact with the bound peptide) is used as task-feature. Clearly, the more similar the pseudo sequences are the more similar we expect the tasks to be. Furthermore, [8] considered several combinations of kernels for K^{all} and K^{pep}. The best performing combination employed a polynomial kernel on top of a string kernel of degree $d = 1$ for both, K^{all} and K^{pep}.

We now aim at improving the above multitask kernel K^{MT} as follows. First, we introduce additional parameters, that allow the specialization of the trade-off between the in-domain kernel components (i.e. $s = t$) and the out-of-domain kernel components (i.e. $s \neq t$) dependent on the task.

While K^{MT} is captures how closely related two tasks are, according to some task kernel K^{all}, it does not take into account how well information from the other tasks boosts performance. Clearly, transferring information from other tasks will become increasingly relevant if only little training data is available. If there is an abundance of training data for a particular task, it is most likely sufficient to set a stronger focus on in-domain data.

The above leads us to the following kernel formulation, which introduces a new multitask kernel composed of a linear combination of two multitask kernels with two mixing coefficients $\beta_{s,1}$ and $\beta_{s,2}$ that have to be adjusted for each task s independently. Details on how the $\beta_{s,k}$ are tuned are given in the following section.

$$\begin{aligned} K^{\text{MT-WD}}((x,s),(z,t)) = & \beta_{s,1} K^{\mathrm{WD}}(s,t) \cdot K^{\mathrm{WD}}(x,z) + \\ & \beta_{s,2} K^{\mathrm{dirac}}(s,t) \cdot K^{\mathrm{WD}}(x,z) \end{aligned} \tag{10}$$

Finally, by combining both lines of work, we propose a multitask kernel that uses the enhanced WD Kernel $K^{\text{WD-RBF}}$ (see Equation 6) from the previous section to compute the similarity between instances. We arrive at the following formulation:

$$\begin{aligned} K^{\text{MT-WD-RBF}}((x,s),(z,t)) = & \beta_{s,1} K^{\mathrm{WD}}(s,t) \cdot K^{\text{WD-RBF}}(x,z) + \\ & \beta_{s,2} K^{\mathrm{dirac}}(s,t) \cdot K^{\text{WD-RBF}}(x,z) \end{aligned} \tag{11}$$

In summary, the new kernel formulation consists of three parts. First, we formulate the kernel as a combination of a task specific component and a multitask kernel component. Second, we introduce task-specific parameters that can be tuned for each task independently. Third, we combine the previous two ideas with the novel WD-RBF kernel.

4 Fine Tuning the Kernel with Multiple Kernel Learning

We propose to use Multiple Kernel Learning (MKL) [9] to learn the weights $\beta_{s,k}$ of the individual components (see Equation 11) along with the respective classifiers. In particular, we employ a variant of MKL, which was shown to work well in the domain of computer vision [5]. Here, the setup is slightly different from standard MKL, as we first obtain one classifier f_i for each kernel K_i (i.e. $f_i(x) = \sum_j \alpha_j y_j K_i(x, x_j)$) and then find an optimal linear combination of the learned functions in a second step (i.e. $f(x,s) = \sum_i \beta_{s,i} f_i(x)$). In [5], the authors propose to use LPBoost for the combination of classifiers. However, LPBoost yields a sparse solution in terms of kernel weights, which is not what we are interested in. Therefore, we propose a formulation based on the nu-SVM [19] to combine the classifiers f_i.

$$\min_{\boldsymbol{\beta},\boldsymbol{\xi},\rho} \frac{1}{2}\|\boldsymbol{\beta} - \mathbf{1}\|^2 + \sum_{i=1}^{N} \xi_i - \rho\nu \tag{12}$$

$$\begin{aligned} \text{s.t.} \quad & y_i \sum_{j=1}^{M} \beta_j f_j(\boldsymbol{x}_i) + \xi_i \geq \rho \ \forall i \in [1, .., N], \\ & \beta_i \geq 0 \ \forall i \in [1, .., M] \\ & \xi_i \geq 0 \ \forall i \in [1, .., N] \end{aligned}$$

From preliminary experiments, we observed that $\beta_{s,k} = 1 \ \forall s \forall k$ often yields a good solution. We use this as prior knowledge by regularizing the parameter vector $\boldsymbol{\beta}$ to be close to a vector of ones $\mathbf{1}$. Intuitively speaking, only the training error measured by the loss term will cause the $\beta_{s,k}$ to differ from 1.

For the training procedure, the training data is split into two parts. The first part containing $\frac{3}{4}$ of training examples is used to obtain the initial f_i. Subsequently, the second part of the training data is used in the loss term of Equation (12), which is solved for each task s individually. After having obtained the β, we retrain the f_i on the entire training data set and use the determined β for the final linear combination $f(x,s) = \sum_{i=1}^{2} \beta_{s,i} f_i(x)$.

5 Experimental Methods

Data. The IEDB benchmark data set from Peters *et al.* [12] contains quantitative binding data (IC_{50} values) for various MHC alleles, including 35 human MHC alleles. Splits for a 5-fold cross-validation are given. We evaluate the performance of the proposed methods on a subset of this data set: binding data of nonameric peptides with respect to human MHC. Peptides with IC_{50} values greater than 500 nM were considered non-binders, all others binders.

Amino acid descriptors. A wide range of physico-chemical and other descriptors of AAs have been published. Within this work we use encode each AA by 20 descriptors corresponding to the respective entries of the Blosum50 substitution matrix [6].

Performance evaluation procedure. For performance evaluation we employ a two times nested 5-fold cross-validation, i.e. two nested cross-validation loops. The inner loop is used for model selection (kernel and regularization parameters) and the outer loop for performance estimation. Performance is measured by averaging the area under the ROC curve (auROC).

Learning curve analysis. To assess the performance dependence on the amount of training data, WD kernel and WD-RBF kernel performances were analyzed on allele A*0201 in 100 cross-validation runs to average over different data splits to reduce random fluctuations of the performance values. In each run, 30% of the available data was used for testing. From the remaining data, training sets of different sizes (20, 31, 50, 80, 128, 204, 324, 516, 822, 1,308) were selected randomly.

Performance analysis of the improved WD kernel. Performances of the WD and the WD-RBF kernel were analyzed on all 35 human MHC alleles contained in the IEDB benchmark.

Performance analysis of the multitask kernel approach. Performances of three multitask learning approaches using a) the WD kernel, b) the WD-RBF kernel, and c) the WD-RBF kernel with an additional optimization step were also analyzed on all 35 human MHC alleles contained in the IEDB benchmark.

SVM computations. We used the freely available large scale machine learning toolbox Shogun [21] for all SVM computations. All used kernels are implemented as part of the toolbox and will be part of Shogun-0.9.3.

Results and Discussion

The main goal of this work is to present novel ideas for kernel-based MHC-I binding prediction, namely an enhanced string kernel [23] and a refined model for multitask learning.

Improved WD Kernel

The more data is available, the easier it will be to infer the relation of the AAs from the sequences in the training data alone. Therefore, the incorporation of additional information can be expected to especially improve prediction accuracy in cases where less training data is available. We chose the allele with the highest number of peptides, A*0201, to perform a learning curve analysis for WD and WD-RBF. Mean auROCs with confidence intervals ($\sigma/\sqrt{n}$) over 100 cross-validation runs are shown in Figure 2. It can clearly be seen, that the fewer examples are available for learning, the stronger is the improvement of the WD-RBF over the WD kernel.

In a more comprehensive comparison, we assessed the performance of WD and WD-RBF kernels on all 35 human MHC alleles from the IEDB benchmark. WD-RBF outperforms WD for 24 alleles (Fig. 3). This is significant with respect to the binomial distribution: Assuming equal performance of WD and WD-RBF, the probability of WD-RBF outperforming WD 24 out of 35 times is ≈ 0.01.

Improved Multitask Learning Kernel

From the results in Figure 4, we can make several important observations. First, in accordance with the results of [7], we clearly see that multitask learning *MTL (WD)* greatly improves performance compared to learning individual models *Plain (WD)*. Second, we observe a slightly improved performance of Plain, when using the WD-RBF instead of the WD, which is consistent with the results from the previous section. In accordance with Figure 2, improvements using the new kernel are rather small as this dataset contains relatively many examples. Third, Figure 4 shows that employing the enhanced multitask Kernel *MTL*

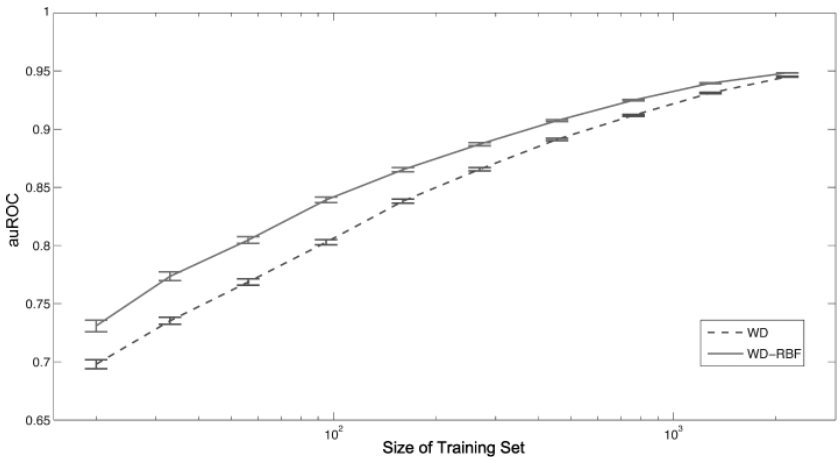


Fig. 2. Learning curve analysis on MHC allele A*0201. Shown are areas under the ROC curves averaged over 100 different test splits (30%) and for increasing numbers of training examples (up to 70%). The training part was used for training and model selection using 5-fold cross-validation.

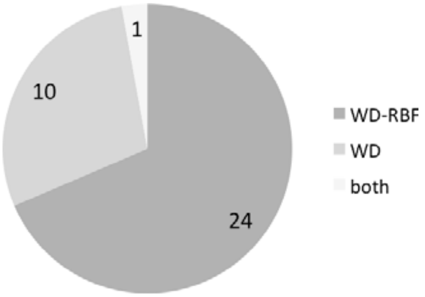


Fig. 3. Performance of WD and WD-RBF kernels on human MHC alleles from the IEDB benchmark data set: The pie chart displays the number of alleles for which the WD (green) and the WD-RBF (red) performed best, respectively, and the number of alleles for which they performed equally (blue).

(WD-RBF) introduced in Equation (11) improves performance compared to the regular multitask learning kernel using the WD kernel. Note, that here, the $\beta_{s,k}$ (see Equation 11) are all set to $\beta_{s,k} = 1$. Lastly, we observe that the tuning the $\beta_{s,k}$ using Equation 12 further improves performance up to $auROC = 0.909$, leaving us with the best performing method in our experiments, which slightly outperforms the method presented in [7], who reported $auROC = 0.903$ for this dataset.

We would like to point out that while the improvement over this previous method is rather small (0.6% auROC), the ideas presented in this paper have the potential to contribute to greater improvements for two reasons. First, [7] used

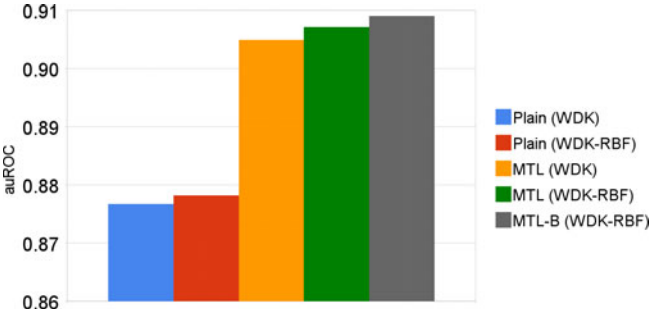


Fig. 4. Performance (averaged over alleles) measured on the IEDB benchmark data set for several methods. In *Plain (WD)/(WD-RBF)* classifiers are trained individually for each task using the WD kernel, or the WD-RBF, respectively. *MTL (WD)* employs a multitask kernel based on the WD, *MTL (WD-RBF)* compares instances using the WD-RBF and *MTL-B (WD-RBF)* employs an additional optimization step (see Equation 12) to fine tune kernel components.

a different *base* kernel. Finding out, whether using this kernel as starting point to our proposed improvements further boosts performance is subject to future experiments. Second, the formulation presented in Equation 12 is extensible to an arbitrary number of kernel components. With more insight into the problem domain, it might be possible to carefully engineer a multitask kernel with more than two meaningful components, which could then be tuned using the proposed formulation.

6 Conclusion

We have proposed two approaches to improve kernel-based Machine Learning methods for MHC class I binding prediction. First, a modification of the Weighted Degree string kernel that allows for the incorporation of amino acid properties. Second, we present an improved multitask learning approach based on a new multitask kernel. Finally, we combine these two approaches, which gives rise to further improvements.

Due to their high dimensional feature space, string kernels require a sufficient number of examples during training to learn relationships between amino acids. Standard kernels employing physico-chemical descriptors of amino acids, on the other hand, cannot exploit the sequential structure of the input sequences and implicitly generate many features, numerous of which will be biologically implausible. Here, one also needs many examples to learn the subset of features that is needed for accurate discrimination. The lack of training data for a large fraction of all known MHC class I alleles, however, calls for approaches that perform well even when training data is scarce. We could show, that incorporation of physico-chemical amino acid descriptors into the Weighted Degree kernel yields significant improvements in the prediction of MHC-binding peptides. This improvement is particularly strong when data is less abundant.

We confirmed that multitask learning methods are beneficial for MHC class I binding prediction. Furthermore, we presented an enhanced multitask kernel that incorporates the improved WD kernel and that has additional hyper-parameters, which are in turn tuned using a variant of the nu-SVM.

Our results show that incorporation of prior knowledge of amino acid properties as well as a sophisticated approach to fine tuning the multitask kernel yields improvements in kernel-based MHC-I binding prediction. While this work focused on the classification into binders and non-binders, the proposed methods show promise also for the quantitative prediction of peptide/MHC class I binding affinity.

Acknowledgments. This work was partly supported by Deutsche Forschungsgemeinschaft (SFB 685, project B1).

References

1. Adams, H.P., Koziol, J.A.: Prediction of binding to MHC class I molecules. Journal of Immunological Methods 185(2), 181–190 (1995)
2. Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E.: The protein data bank. Nucleic Acids Research 28, 235–242 (2000)
3. Dönnes, P., Elofsson, A.: Prediction of MHC class I binding peptides, using SVMHC. BMC Bioinformatics 3, 25 (2002)
4. Evgeniou, T., Pontil, M.: Regularized multi–task learning. In: Kim, W., Kohavi, R., Gehrke, J., DuMouchel, W. (eds.) Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA, August 22-25, pp. 109–117. ACM, New York (2004)
5. Gehler, P., Nowozin, S.: Infinite kernel learning. In: NIPS Workshop on Kernel Learning: Automatic Selection of Optimal Kernels (2008)
6. Henikoff, S., Henikoff, J.G.: Amino acid substitution matrices from protein blocks. Proceedings of the National Academy of Sciences of the United States of America 89(22), 10915–10919 (1992)
7. Jacob, L., Bach, F., Vert, J.P.: Clustered Multi-Task Learning: A Convex Formulation. In: NIPS, pp. 745–752. MIT Press, Cambridge (2009)
8. Jacob, L., Vert, J.P.: Efficient peptide-MHC-I binding prediction for alleles with few known binders. Bioinformatics 24(3), 358 (2008)
9. Kloft, M., Brefeld, U., Sonnenburg, S., Zien, A., Laskov, P., Müller, K.R.: Efficient and accurate LP-norm MKL. In: Advances in Neural Information Processing Systems, vol. 22 (2009)
10. Kuang, R., Ie, E., Wang, K., Wang, K., Siddiqi, M., Freund, Y., Leslie, C.: Profile-based string kernels for remote homology detection and motif extraction. In: Proceedings IEEE Computational Systems Bioinformatics Conference (2004)
11. Moll, A., Hildebrandt, A., Lenhof, H., Kohlbacher, O.: BALLView: an object-oriented molecular visualization and modeling framework. J. Comput. Aided Mol. Des. 19(11), 791–800 (2005)
12. Peters, B., Bui, H.H., Frankild, S., Nielsen, M., Lundegaard, C., Kostem, E., Basch, D., Lamberth, K., Harndahl, M., Fleri, W., Wilson, S.S., Sidney, J., Lund, O., Buus, S., Sette, A.: A Community Resource Benchmarking Predictions of Peptide Binding to MHC-I Molecules. PLoS Comput. Biol. 2(6), e65 (2006)

13. Pfeifer, N., Kohlbacher, O.: Multiple Instance Learning Allows MHC Class II Epitope Predictions Across Alleles. In: Crandall, K.A., Lagergren, J. (eds.) WABI 2008. LNCS (LNBI), vol. 5251, pp. 210–221. Springer, Heidelberg (2008)
14. Rammensee, H., Bachmann, J., Emmerich, N.P., Bachor, O.A., Stevanovic, S.: SYFPEITHI: Database for MHC ligands and peptide motifs. Immunogenetics 50, 213–219 (1999)
15. Rätsch, G., Sonnenburg, S.: Accurate Splice Site Detection for *Caenorhabditis elegans*. In: Schölkopf, B., Vert, K.T. (eds.) Kernel Methods in Computational Biology, pp. 277–298. MIT Press, Cambridge (2004)
16. Rätsch, G., Sonnenburg, S., Srinivasan, J., Witte, H., Müller, K.R., Sommer, R.J., Schölkopf, B.: Improving the *Caenorhabditis elegans* genome annotation using machine learning. PLoS Comput. Biol. 3(2), e20 (2007)
17. Reche, P.A., Glutting, J.P., Reinherz, E.L.: Prediction of MHC class I binding peptides using profile motifs. Hum. Immunol. 63(9), 701–709 (2002)
18. Schölkopf, B., Burges, C., Smola, A. (eds.): Advances in Kernel Methods: Support Vector Learning. MIT Press, Cambridge (1999)
19. Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L.: New support vector algorithms. Neural Computation 12(5), 1207–1245 (2000)
20. Schweikert, G., Zien, A., Zeller, G., Behr, J., Dieterich, C., Ong, C.S., Philips, P., De Bona, F., Hartmann, L., Bohlen, A., Krüger, N., Sonnenburg, S., Rätsch, G.: mGene: accurate SVM-based gene finding with an application to nematode genomes. Genome Res. 19(11), 2133–2143 (2009)
21. Sonnenburg, S., Rätsch, G., Schäfer, C., Schölkopf, B.: Large Scale Multiple Kernel Learning. Journal of Machine Learning Research 7, 1531–1565 (2006)
22. Toussaint, N.C., Kohlbacher, O.: Towards in silico design of epitope-based vaccines. Expert Opinion on Drug Discovery 4(10) (2009)
23. Toussaint, N.C., Widmer, C., Kohlbacher, O., Rätsch, G.: Exploiting physicochemical properties in string kernels. BMC Bioinformatics (submitted, 2010)
24. Tung, C.-W., Ho, S.-Y.: POPI: predicting immunogenicity of MHC class I binding peptides by mining informative physicochemical properties. Bioinformatics 23(8), 942–949 (2007)

Part II

Unsupervised Learning Methods for Biological Sequences

SIMCOMP: A Hybrid Soft Clustering of Metagenome Reads

Shruthi Prabhakara* and Raj Acharya

Department of Computer Science and Engineering
Pennsylvania State University, University Park, State College, PA 16801
{sap263,acharya}@cse.psu.edu

Abstract. A major challenge facing metagenomics is the development of tools for the characterization of functional and taxonomic content of vast amounts of short metagenome reads. In this paper, we present a two pass semi-supervised algorithm, SimComp, for soft clustering of short metagenome reads, that is a hybrid of comparative and composition based methods. In the first pass, a comparative analysis of the metagenome reads against BLASTx extracts the reference sequences from within the metagenome to form an initial set of seeded clusters. Those reads that have a significant match to the database are clustered by their phylogenetic provenance. In the second pass, the remaining fraction of reads are characterized by their species-specific composition based characteristics. SimComp groups the reads into overlapping clusters, each with its read leader. We make no assumptions about the taxonomic distribution of the dataset. The overlap between the clusters elegantly handles the challenges posed by the nature of the metagenomic data. The resulting cluster leaders can be used as an accurate estimate of the phylogenetic composition of the metagenomic dataset. Our method enriches the dataset into a small number of clusters, while accurately assigning fragments as small as 100 base pairs.

1 Introduction

Metagenomics is defined as the study of genomic content of microbial communities in their natural environments, bypassing the need for isolation and laboratory cultivation of individual species[1]. Its importance arises from the fact that over 99% of the species yet to be discovered are resistant to cultivation[2]. Metagenomics promises to enable scientists to study the full diversity of the microbial world, their functions and evolution, in their natural environments.

Metagenomics projects collect DNA from environments that are characterized by large disparity in sequence coverage and abundance of species distribution. Sequencing technologies are used to survey the metagenomic content. The recent ultra-high throughput sequencing technologies [3] produce relatively short reads, 25-400 base pairs(bp), and enormous datasets, thereby creating new computational challenges for metagenomics. It is critical that we develop fast and

* Corresponding author.

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 113–124, 2010.

accurate tools for assembling and characterizing the phylogenetic provenance and the relative abundance of different species in a metagenomic sample. Clustering of metagenome reads is one such tool that provides deeper insight into the structure of the community and hence, can be used to model the ecological and population parameters. This pre-processing step can lead to faster and more robust assembly by reducing the search space[14].

2 Related Work

Methods for clustering reads proposed so far in literature can be categorized into two main approaches; comparative(or similarity) and composition based. Comparative based methods align metagenomic sequences to close phylogenetic neighbors in existing databases and hence depend on the availability of closely related genomes in the database[7,6,11]. Such methods fail to find any homologs for new families. Composition based methods, on the other hand, distinguish between clades by using intrinsic features of reads such as oligomer frequencies[10,12,13], codon usage preferences[17] or GC content[16]. The strength of this approach is that no reference database is required. However, oligomer composition of reads shorter than 1 kbp carry insufficient signal to be able to differentiate between species. Composition based clustering of metagenome reads complements the comparative analysis[12].

The last decade has seen an explosion in the computational methods developed to analyze metagenomic data. A number of methods for classifying(as opposed to clustering) metagenome reads into taxon-specific bins have been proposed in literature. Phylopythia[10] is a supervised composition based classification method that trains a support vector machine to classify sequences of length greater than 1 kbp. Phymm uses interpolated markov models to characterize variable length DNA sequences into their phylogenetic groups[12]. Its accuracy of assignment drops drastically for short reads and reads from unknown species. Nasser et al.[14] demonstrated that a k-means based fuzzy classifier, trained using a maximal order markov chain, can separate 1kbp reads with a high accuracy at phylum level. All the above supervised methods depend on the availability of reference data for training. These methods assume the prior knowledge of the number of classes. A metagenomic dataset may contain reads from unexplored phyla which cannot be labeled into one of the existing classes.

Li et al. propose a composition based leader clustering algorithm that clusters highly homologous sequences in order to condense a large database[9]. More recently, Chan et al. developed a semi-supervised seeded growing self-organizing map to cluster metagenomic sequences[18]. It extracts 8-13 kbp of flanking sequences of highly conserved 16S rRNA from the metagenome and uses them as seeds to assign the remaining reads using composition based clustering. CompostBin uses weighted PCA to project the DNA composition data into lower-dimensional space, and then uses the normalized cut clustering to classify reads into taxon-specific bins[20]. Likely-Bin is an unsupervised method for binning short reads by taxonomy on the basis of their k-mer distributions[21].

MEGAN, a metagenome analysis software system [11], on the other hand, uses sequence homology to assign reads to common ancestors based on best match as given by BLAST(Basic Local Alignment Search Tool)[19]. As most of the extant databases are highly biased in their representation of true diversity, methods such as MEGAN fail to find any homologs for new families. Most metagenomic analysis methods until now have been relatively inaccurate in classifying reads as short as 100 base pairs.

Increased amounts of polymorphism and horizontal gene transfer in metagenome reads leads to conflicts in assembly and taxonomic analysis. Reads from closely related species will most likely have homologous sequences shared between clusters that fuzzify the cluster boundaries[18]. Another characteristic of these datasets is the incomplete and fragmentary nature of the metagenome reads that reduces the quality of annotation. However, clipping low quality reads such as chimeras can exclude potentially useful sequences. Hence, in light of the new data, we need to adapt the traditional approaches to metagenome analysis. Overlapping clusters generated by a soft clustering algorithm such as the one proposed in this paper elegantly handle the problems associated with the nature of metagenomic data while providing tolerance for the noise due to errors in sequencing and fragmentation. The soft boundaries between clusters provide the flexibility to capture the misplacements of reads due to polymorphism or over representation of conserved regions, thereby providing interesting insights into the data.

Our work is inspired by the works of Dalevi et al.[6] and Folino et al.[7]. In [6], the authors propose a method for clustering reads based on a set of proteins, called proxygenes. The protein hits are obtained by BLASTx (specialized nucleotide-protein BLAST) of the reads against a reference proteome database. Their work is extended in [7], where a method based on weighted proteins is used to cluster the reads, resulting in overlapping clusters, each represented by a proxygene. The underlying basis of the above methods is that a high sequence similarity between the read and the proxygene implies phylogenetic proximity of the organisms from which they originated [6]. Consequently, the taxonomic annotation of the proxygene can be used in assessing that of the reads in the cluster. Both the methods use the comparative approach and hence rely on the use of a reference database that contains closely related genomes. However, in a typical metagenome dataset, majority of the reads may exhibit no similarity to any known sequence in the database. In such a scenario, these methods will fail to assign these reads to any cluster.

In this paper, we propose a two pass semi-supervised algorithm for soft clustering of short metagenome reads. We call our method SimComp; a hybrid of similarity and composition based methods. The objective of our method is to enrich the dataset into a small number of clusters such that reads within a cluster are phylogenetically closer than reads from different clusters. Each cluster is defined by a core consisting of reads that definitely belong to the cluster and a fringe that has reads which may overlap with other clusters. We make no assumptions about the taxonomic distribution of the metagenome dataset. SimComp makes use of a reference database, however is not dependent on it.

In the first pass, a comparative analysis of the metagenome reads against an existing database, using BLASTx, extracts reference sequences from within the dataset to form an initial set of seeded clusters. Reads that have a significant match to the database are clustered by their phylogenetic provenance. In the second pass, the global clade-specific characteristics(e.g. oligomer frequency) are used to cluster the remaining reads by a soft leader clustering algorithm described in [1]. Our algorithm groups the reads into overlapping clusters, each with its read leader. The fringes of the clusters accomodate the ambiguity associated with reads in the dataset. It automatically performs the selection of the number of clusters. Essentially, the comparative analysis of reads avails apriori biological knowledge in existing protein database to form initial set of seeded clusters.Then, the composition based characterization of remaining fraction of reads, thereby facilitating a means of exploring novel species.

3 An Overview of Methods and Algorithm

SimComp is based on the Adaptive Rough Fuzzy Leader Clustering presented by Asharaf et al.[8]. The authors use rough set theory to define the clusters. Each cluster has a core(lower bound) and a fringe(upper bound) and is represented by a read leader. The core contains all the reads that definitely belong to the cluster. Reads in the core are mutually exclusive between the clusters. There can be an overlap in the fringes of two or more clusters.

3.1 Comparative Clustering

In the comparative pass of the algorithm, as in [7,6], we associate a list of protein hits with each read, identified by BLASTx. Each hit consists of one protein, two score values called bits and identities which describe the significance of read-protein alignment, and a confidence value called E-value which describes the likelihood that the sequence will occur in the database by chance. We further use the measure defined in [7] (explained in the Appendix) for assigning weights to the each of the proteins, such that proteins that cover more reads are assigned smaller weights. Proteins that are below a predefined protein threshold form the proxygenes, the rest are discarded. The proxygenes are clustered with the corresponding best hit reads(as identified by BLASTx). For each cluster thus formed, the most representative read is chosen as the leader(seed of a cluster).

3.2 Composition Based Clustering

The reads remaining after the first pass are clustered using the soft leader clustering algorithm based on sequence composition. In this pass, each unclustered read is compared with the existing read leaders. The similarity between the read and the leaders along with the sequence thresholds determines whether the read gets added to the core of some cluster or fringes of one or more clusters, or the read itself gets added as a leader. The steps in SimComp are outlined below.

3.3 Definitions

Cluster. Each cluster consists of a read leader, representative of the set of reads in the cluster. A cluster is defined by the following parameters:

- Protein threshold (PT): Proteins with weight below the threshold form proxygenes. Each proxygene is representative of a cluster with the corresponding reads(as identified by BLASTx). Rest of the proteins are discarded. The weight assigned to a protein is measured by two score values, i.e. bits and identities, and a confidence value called E-value[7].
- User defined core and fringe sequence similarity threshold for clusters (RT_C and RT_F): If the similarity between the read and its nearest leader is greater than RT_C, the read is added to the core of a cluster. Otherwise, if the similarity between the read and the corresponding cluster leaders is greater than RT_F, the read is added to the fringes of one or more clusters.

Sequence similarity. Each sequence is represented by a vector of oligomer frequencies, $v = (f_1, f_2 ... f_q)$; where for each oligomer of length n, $O = (o_1, o_2 ... o_q)$ is the set of all possible oligomers, f_i is the frequency of oligomer pattern o_i in the read, q is the number of oligomer patterns of length n possible, i.e. 4^n. Each vector is normalized relative to the length of the sequence. $S(x, y)$ gives the similarity between read x and leader y. We define sequence similarity as the number of fixed length oligomers shared between x and y.

Fuzzy membership. U_{ik} is the fuzzy membership of the read r_i in a cluster represented by Leader L_k.

$$U_{ik} = \sum_{j=1}^{N} \frac{S(r_i, L_k)}{S(r_i, L_j)} \tag{1}$$

3.4 SIMCOMP : Outline of the Algorithm

The algorithm proceeds in two passes. Let $R = (r_1, r_2, ... r_n)$, be the set of all reads and N be the number of clusters at any point in the algorithm.

I. Comparative Clustering: In the first pass, metagenome reads are grouped into clusters based on similarity of the reads to the proteins in the reference database.

1. Extract all proteins that R has hits to(by BLASTx).
2. Assign weights to all the proteins based on equation described in [7] (see Appendix). Proteins with weight below PT form proxygenes.
3. Each proxygene, along with the corresponding best hit reads (identified by BLASTx) form a cluster.
4. For each of the clusters, find a read leader that is most representative of the reads in the cluster, i.e. one whose sum of sequence similarity from all the other reads in the cluster is maximum.

II. Composition Based Clustering: In the second pass, we use the similarity measure based on oligomer frequency(defined above) to cluster the remaining reads.

1. All the reads from the original dataset that have not yet been clustered form the remaining read set. For each read in the remaining read set, compare the read with the existing read leaders. Depending on the value of RT_C, RT_F and sequence similarity between the read and the leaders, one of the three cases can arise for assignment of the current read:
   (a) It gets added to the core of a cluster. The current read gets added to the core of a cluster represented by leader L_p, if $\max(S(r_i, L_k)/k = 1...N) = D_{ip}$ and $D_{ip} > RT_C$.
   (b) It gets added to the fringes of one or more clusters. r_i falls into the fringes of all the clusters L_p for which $S(r_i, L_p) > RT_F$ and $S(r_i, L_p) < RT_C$.
   (c) Otherwise, r_i gets added as leader since it is outside the region defined by any of the existing clusters.

4 Results

We implemented our algorithm in Matlab. All experiments were run on an IBM X3550 server with 8GB memory. We tested our method on the simulated metagenome datasets M1, M2 and M3, introduced in [6], each at a coverage of 0.1X. These datasets were sequenced at Joint Genome Institute using the 454 pyrosequencing platform that produces ∼100 bp reads. We present results from experiments on M1 dataset only due to constraints in space. The characterization of reads at the taxonomic level of an organism for M1 is as shown in Fig 1. We used the default parameters of BLASTx, and NR[15] (Non-Redundant) protein sequence database as our reference. We have conducted experiments for varying values of user-defined thresholds(RT_C, RT_F) and lengths of oligomers. Based on the evaluation of our method on M2 and M3, we observed that proteins with weight below the 1^{st} percentile cover all the taxonomies that reads belong to. Therefore, we selected the 1^{st} percentile of weight as our protein threshold. The most time consuming component of SimComp is generating the BLASTx output. Once this output has been generated, the algorithm performs a single pass over the BLASTx output and the dataset to cluster the reads and hence is very efficient.

4.1 Accuracy across Taxonomic Ranks

In this paper, we use two measures to evaluate the effectiveness of our method: Mode Cluster Purity and Leader Cluster Purity. Mode Cluster Purity is defined as the maximum fraction of reads in a cluster belonging to the same taxon[7]. We define Leader Cluster Purity as the fraction of elements in the cluster belonging to the same taxon as the read leader. This measure determines how well our algorithm models the problem of classifying reads from species that have never been seen before. Depending on the elements of the cluster that we evaluate on, cluster purity can be further divided into core cluster purity(all the reads in the core of the cluster) and total cluster purity(all the reads in the cluster). In evaluating both the measures, we take into account only the non-singleton clusters, as a singleton cluster has a cluster purity of 1.

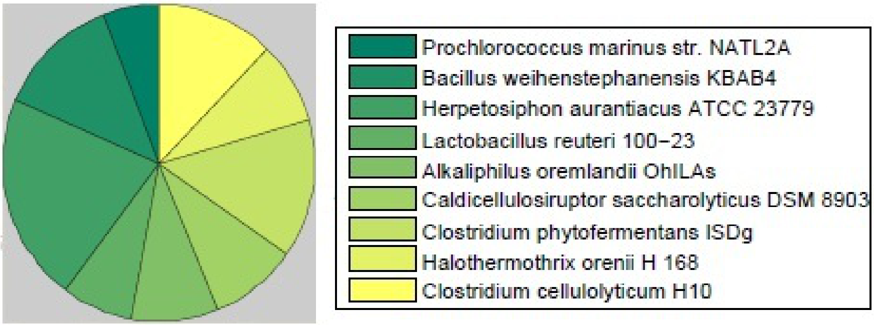


Fig. 1. Organism level characterization of M1 dataset

In Fig 2, we plot the taxonomic distribution of reads in M1 at phylum, class, order and family level($RT_C = 15$ and $RT_F = 12$ and length of oligomer = 6) as predicted by our algorithm. To measure the taxonomic distribution, all the reads in the cluster are assigned the same taxa as the read leader. Our method yields satisfactory results at all ranks. Hence, leaders of the clusters can be used as an accurate estimate of the phylogenetic composition of the metagenome. In [6,7], only those reads that have significant hits in the BLASTx output are selected for further clustering, the remaining reads are discarded. As opposed to this, in our method, we cluster all the reads in the dataset, even if no significant hits to the reference database are obtained. In Fig 3, we have plotted three measures for dataset M1 across all taxonomic ranks. By definition, mode cluster purity is greater than or equal to leader cluster purity. From the plot, we conclude that the cluster purity of the core is higher than that of the entire cluster at all ranks. This asserts our algorithms ability to filter out low quality reads into the fringe of a cluster.

4.2 Length of Oligomer

Oligomer frequency of genomes has been shown to reflect clade-specific characteristics and thus form a genome signature[4]. Teeling et al.[5] have shown that tetranucleotide frequency has a higher discriminatory power than GC content for phylogenetic grouping of reads. We have evaluated the accuracy of assignment of reads to clusters for a range of oligomers varying from trimers to hexamers. Fig 4 shows the plot of percentage of non-singleton clusters with purity values in the range [0.1,1] for varying lengths of oligomer. From our experiments, we conclude that hexamers have the best discriminatory power for clades at higher taxonomic ranks. With reads as small as 100 bp, not many reads cross that high a similarity threshold for hexamers. This explains the increase in number of singleton clusters with the increase in read threshold.

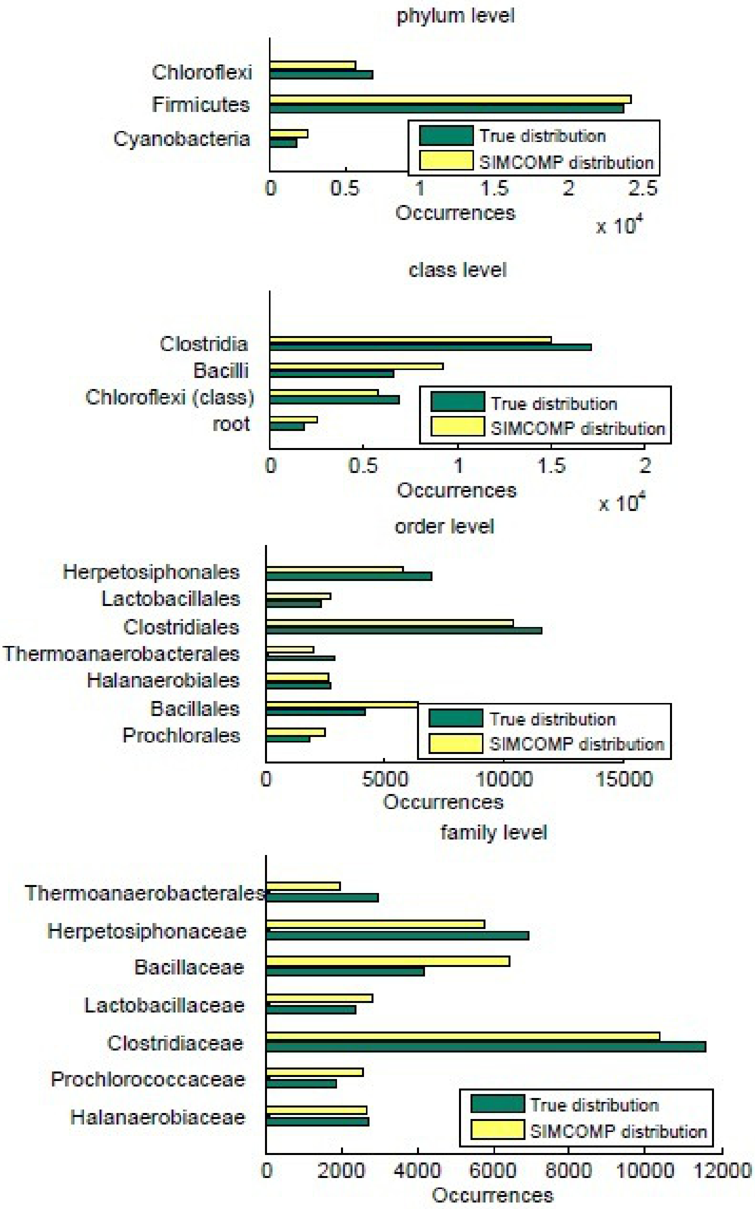


Fig. 2. Taxonomic Distribution Across Ranks (Phylum, Class, Order, Family

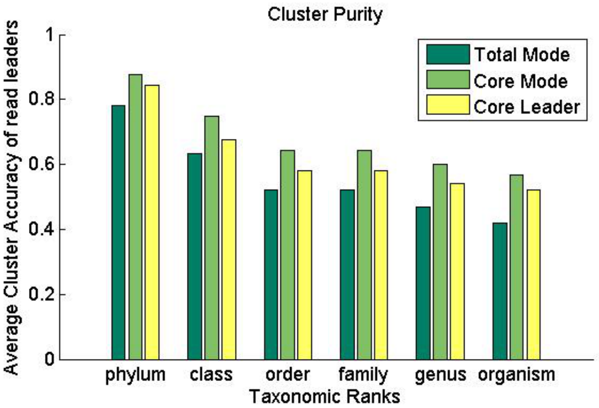


Fig. 3. Average cluster purity across taxonomic ranks for ($RT_C = 15$ and $RT_F = 12$ and length of oligomer = 6, Number of Clusters = 2430)

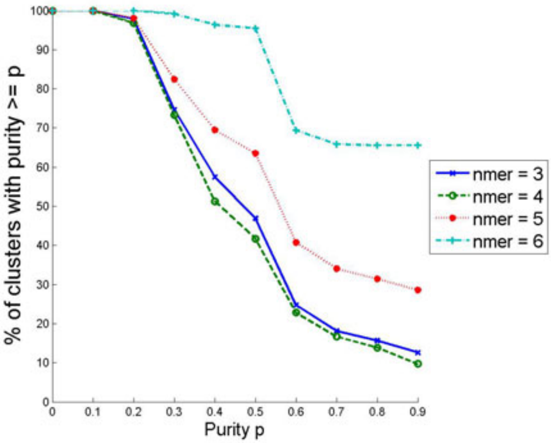


Fig. 4. Plot of percentage of non-singleton clusters for different values of purity with $RT_C = 25$ and $RT_F = 22$ and varying values of oligomers

4.3 Read Threshold

In our method, sequence similarity between two reads is measured as a function of number of fixed length oligomers shared between the two reads. A read is added to the core of an existing cluster only if the read similarity between the read and the cluster leader is above a certain threshold. Fig 5 plots the mode cluster purity for different values of read thresholds. The curve for $RT_C = 25$ clearly dominates the others. This is justified as clusters with large read thresholds are smaller in size and hence are likely to have a high purity. Table.1 summarizes the results for a fixed oligomer length of 6 and varying read thresholds. Cluster purity increases with the increase in read thresholds, for the reasons cited above.

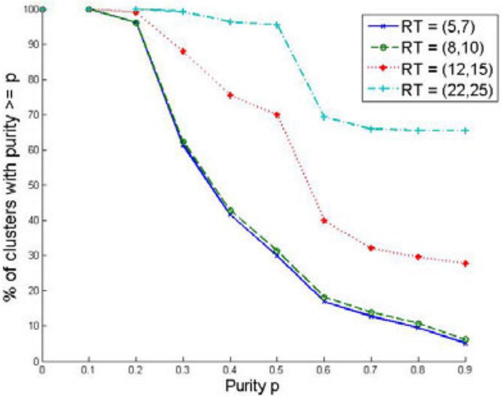


Fig. 5. Plot of percentage of non-singleton clusters for different values of purity with oligomer lenght = 6 and varying values of Read Threshold (Core, Fringe)

Table 1. Summary of the results of experiments for oligomer length = 6 and varying Read Thresholds

RT_C	10	15	20
RT_F	8	12	17
Number of Clusters	1482	2430	14250
Maximum size of clusters	320	415	288
Number of singleton clusters	6	67	5865
Reduction factor	0.042	0.068	0.4
Mode Cluster Purity at Phylum level	79.93	88.14	96.95
Mode Cluster Purity at Organism level	40.88	61.75	88.41

5 Conclusion

In this paper, we proposed SimComp, a soft clustering method that allows complete and accurate characterization of short metagenome reads that come from a spectrum of known and unknown species. We clustered a simulated dataset using a hybrid of comparative and composition based method. The overlap between the clusters accomodates the ambiguity associated with metegenomic data. It does not require assembled contigs or training on a reference set, nor does it make any assumptions on the number of species or the nature of the dataset.

The oligomer composition of reads as short as 100 bp does not provide sufficient signal to differentiate between species. For best results, we would like to test our algorithm on metagenome datasets with larger read length. Phenomena such as polymorphism and horizontal gene transfer can complicate phylogenetic clustering. As proposed in this paper, the soft boundary between clusters has the ability to capture such misplacements providing interesting insights into the data. We believe soft clustering has a promising role in classifying metagenome reads and we wish to investigate its scope in the future.

Acknowledgments. We would like to thank Mavrommatis Konstantinos for providing datasets in [6] and Elena Marchiori and Fabio Gori for providing useful information about their paper [7]. We would also like to thank Piotr Berman for suggesting useful changes to the paper.

References

1. Chen, K., Pachter, L.: Bioinformatics for whole-genome shotgun sequencing of microbial communities. PLoS Comp. Biol., 1–24 (2005)
2. Rappe, M.S., Giovannoni, S.J.: The uncultured microbial majority. Annual Rev. Microbiol., 357–369 (2003)
3. Pop, M., Salzberg, S.L.: Bioinformatics challenges of new sequencing technology. Trends Genet. 24, 142–149 (2008)
4. Karlin, S., Ladunga, I., Blaisdell, B.E.: Heterogeneity of genomes: measures and values. Proc. Natl. Acad. Sci. USA 91, 12837–12841 (1994)
5. Teeling, H., Meyerdierks, A., Bauer, M., Amann, R., Glockner, F.: Application of tetranucleotide frequencies for the assignment of genomic fragments. Environmental Microbiology 6, 938–947 (2004)
6. Dalevi, D., Ivanova, N.N., Mavromatis, K., Hooper, S.D., Szeto, E., Hugenholtz, P., Kyrpides, N.C., Markowitz, V.M.: Annotation of metagenome short reads using proxygenes. Bioinformatics 24(16) (2008)
7. Folino, G., Gori, F., Jetten, M.S., Marchiori, E.: Clustering Metagenome Short Reads Using Weighted Proteins. In: EvoBIO '09: Proceedings of the 7th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics (2009)
8. Asharaf, S., Narasimha Murty, M.: An adaptive rough fuzzy single pass algorithm for clustering large data sets. Pattern Recognition 36(12) (2003)
9. Li, W., Godzik, A.: Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics 22 (2006)
10. McHardy, A.C., Martin, H.G., Tsirigos, A., Hugenholtz, P., Rigoutsos, I.: Accurate phylogenetic classification of variable-length DNA fragments. Nature Methods 4, 63–72 (2007)
11. Huson, D.H., Auch, A.F., Qi, J., Schuster, S.C.: MEGAN analysis of metagenomic data. Genome Res. 17, 377–386 (2007)
12. Brady, A., Salzberg, S.L.: Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models. Nature Methods 1358 (2009)
13. Teeling, H., Waldmann, J., Lombardot, T., Bauer, M., Glockner, F.O.: Tetra: a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in DNA sequences. BMC Bioinformatics 5, 163 (2004)
14. Nasser, S., Breland, A., Harris, F.C., Nicolescu, M.: A fuzzy classifier to taxonomically group DNA fragments within a metagenome. Annual Meeting of the North American Fuzzy Information Processing Society, 1–6 (2008)
15. Non-Redundant Proteome database, ftp://ftp.ncbi.nlm.nih.gov/blast/db
16. Bentley, S.D., Parkhill, J.: Comparative genomic structure of prokaryotes. Annual Review of Genetics 38, 771–792 (2004)
17. Bailly-Bechet, M., Danchin, A., Iqbal, M., Marsili, M., Vergassola, M.: Codon Usage Domains over Bacterial Chromosomes. PLoS Computational Biology 2(4), e37 (2006)

18. Chan, C., Hsu, A., Halgamuge, S., Tang, S.: Binning sequences using very sparse labels within a metagenome. BMC Bioinformatics 9, 215 (2008)
19. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. J. Mol. Biol. 215, 403–410 (1990)
20. Chatterji, S., Yamazaki, I., Bai, Z., Eisen, J.: CompostBin: A DNA composition-based algorithm for binning environmental shotgun reads. In: Vingron, M., Wong, L. (eds.) RECOMB 2008. LNCS (LNBI), vol. 4955, pp. 17–28. Springer, Heidelberg (2008)
21. Kislyuk, A., Bhatnagar, S., Dushoff, J., Weitz, J.S.: Unsupervised statistical clustering of environmental shotgun sequences. BMC Bioinformatics 10, 316 (2009)

Appendix

As in [7], from each hit that BLASTx outputs for a given read r, we extract a 4-dimensional vector $h = (p; S_B; Id; E)$ where p is the matched protein, S_B the bit score, Id the identities score, and E the E-value of that match. For a read r let Hit_r be the sequence, sorted in increasing order of E-values, of its hits. Denote by $r_1, ..., r_m$ the set of reads r with non-empty Hit_r. Let $P = \{p_1, ..., p_n\}$ be the set of proteins occurring in $\cup_{i=1}^{m} Hit_i$ For each protein $p \in P$, the set H_p is defined as:

$$H_p = \{h \in \cup_{i=1}^{m} Hit_i | h(1) = p\} \tag{2}$$

where $h(1)$ denotes the first component of the hit vector h. Thus H_p consists of the selected hits containing p. We use the equation described in [7] to assign weights to the each of the protein hits that BLASTx outputs. Weight of protein p is defined as:

$$w_p = 1 + \lceil \frac{1}{|H_p|} \sum_{h \epsilon H_p} (100 \frac{max\_score - S_B(h)}{max\_score - min\_score} + 100 - Id(h)) \rceil \tag{3}$$

where H_p consists of hits containing p, $S_B(h)$ and $Id(h)$, the bit and identity score of hit h respectively. For further details, we refer the reader to [7].

The Complexity and Application of Syntactic Pattern Recognition Using Finite Inductive Strings

Elijah Myers[1], Paul S. Fisher[1], Keith Irwin[1], Jinsuk Baek[1], and Joao Setubal[2]

[1] Department of Computer Science, Winston-Salem State University, USA
{emyers106,fisherp,irwinke,baekj}@wssu.edu
[2] Viginia Bioinformatics Institute, Virgina Tech Blacksburg, VA 24061, USA
setubal@vbi.vt.edu

Abstract. We describe herein the results of implementing an algorithm for syntactic pattern recognition using the concept of Finite Inductive Sequences (FI). We discuss this idea, and then provide a big O estimate of the time to execute for the algorithms. We then provide some empirical data to support the analysis of the timing. This timing is critical if one wants to process millions of symbols from multiple sequences simultaneously. Lastly, we provide an example of the two FI algorithms applied to actual data taken from a gene and then describe some results as well as the associated data derived from this example.

Keywords: Pattern Recognition, finite induction, syntactic pattern recognition, algorithm complexity.

1 Introduction

Despite the fact that there has been extensive research and development within the pattern recognition topic, new problems continue to emerge that require more efficient revisions of existing techniques and, occasionally, new techniques to solve existent problems. For example, the problems associated with finding motifs [1], [2] are particularly difficult due to mutations, unknown boundaries, etc. While many new problems continue to emerge that could potentially benefit from the use of pattern recognition, but the current effort reported herein is an extension with applications of previous work [3] in reference to the field of bioinformatics, where it is often the case that genetic data is processed for a vast multitude of diverse purposes. Regardless of the purpose of the research, bioinformatics often entails processing genetic data in the form of strings consisting of the symbols A, C, G, and T as well as equivalent protein sequences. This type of string is suitable for syntactic pattern recognition using finite inductive (FI) sequences, but again there are some issues that need to be addressed, and we will address some of them later in this paper. It is the purpose of the FI algorithms [4] to provide a general technique to achieve pattern recognition when comparing finite strings in order to determine a) what patterns exist in the examined strings, and b) whether or not subsequent strings contain similar or identical subsequences in the same form as such exemplar substrings are known by the algorithms.

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 125–136, 2010.

2 Review of FI Algorithms and Theory

The idea [4] is to introduce a 'ruling' as a finite machine that can, when provided a short driving sequence, generate a sequence that is much longer. The 'rules' called *implicants* contained within the ruling come from the processing of a finite sequence of symbols constructed from the designated alphabet. We further stipulate that the choice of any symbol at any particular position depends upon only the symbols at the previous n points. The least such n is called the inductive base (IB) for the sequence. We define an implicant as the pair (w, p) consisting of a word w over the alphabet and a single member p of that alphabet. We also require that w occurs at least once, and whenever w occurs, then it is followed immediately by p. We express this relationship as $w \rightarrow p$, and call w the antecedent and p the consequent. We also assume w is in reduced form: there is no proper terminal segment that is the antecedent of another implicant. We can state the following simple properties:

- For any finite sequence, the IB is the maximum length of the antecedents in the reduced form implicants.
- If an FI sequence has inductive base A and contains b symbols in the alphabet, then the upper bound for the reduced form implicants is b^A.

For purposes of simplicity, we will assume there is a distinguished symbol S that serves as the start symbol for all FI sequences. We also state without proof that if the original implicants (called *prime implicants*) generated from the sequence have inductive bases that differ among themselves, then it is possible to reduce the inductive base b of the implicants to a value $1 \leq \text{IB} < b$.

2.1 Generating and Applying the FI Algorithms

There are two algorithms that make up the FI system. These are called *Factoring* and *Following*. Factoring is the process whereby a storage structure called *Ruling* is generated based upon an a'priori IB, and Following is the process whereby the ruling is applied to unknown patterns.

Example 1: Factoring. Suppose we have a sequence aactgctagt. We append the start symbol and then begin the process of factoring, and we will allow the IB to be as large as necessary to accommodate all of the implicants in one level (called Prime Implicants).

$$\text{Input Sequence: } S\text{aactactagt} \tag{1}$$

Implicants: $S \rightarrow$ a, Sa $\rightarrow$ a, aa $\rightarrow$ c, ac $\rightarrow$ t, ct $\rightarrow$ a, aacta $\rightarrow$ c, tac $\rightarrow$ t, tacta $\rightarrow$ g, and g $\rightarrow$ t

As can be seen from the implicants, the IB is 5, and there are other implicants with IB less than 5, so we can reduce these prime implicants to new implicants with IB say 2. We do so in the following steps:

Step 1: We note that the following implicants meet our new IB value of 2:

$$S \rightarrow \text{a}, S\text{a} \rightarrow \text{a}, \text{aa} \rightarrow \text{c}, \text{ct} \rightarrow \text{a}, \text{ac} \rightarrow \text{t}, \text{g} \rightarrow \text{t} \tag{2}$$

This leaves the symbols in the string from (1) as follows in (3) where the consequents not kept (pushed out) are shown in Level 1:

Level 1	S				t		c		a	g	
Level 0	S	a	a	c	t	a	c	t	a	g	t

(3)

Step 2: We apply the same process of Step 1 to the symbols remaining in Level 1. Level 1 is called the residual for Level 0. This produces the following rules (4) with an empty residual:

$$S \rightarrow \mathrm{t}, \mathrm{t} \rightarrow \mathrm{c}, \mathrm{c} \rightarrow \mathrm{a}, \mathrm{a} \rightarrow \mathrm{g} \tag{4}$$

From (2) and (4) we can now define the ruling with inductive base 2 for the sequence aactactagt with driving sequence S.

Level 1	$S \rightarrow$ t	t $\rightarrow$ c	c $\rightarrow$a	a $\rightarrow$g		
Level 0	$S \rightarrow$ a	Sa $\rightarrow$ a	aa $\rightarrow$ c	ct $\rightarrow$ a	ac $\rightarrow$ t	g $\rightarrow$ t

(5)

Example 1: Following. Suppose we have a new sequence Saactggacattac and we want to process it against our known sequence as represented by the ruling in Example 1.

Step 1: We apply all implicants of (5) in Level 0 to see for the given antecedents if the consequent matches. If it matches, then the consequent is deleted at the end of the processing for this level. The symbols bolded indicate that the symbol is deleted.

S **a a c t** g g a c a t t a c

Step 2: We apply the implicants of (5) in Level 1 to the residual of Step 1.

S g g a c **a** t t a c

This results in the residual string S g g a c t t a c. One cannot say much about the two strings as how they relate to one another, since they do not represent much data; but we can say in general that the two sequences are not very similar to one another. At this stage, we could add the sequence of Example 2 to the ruling, if it was important. In general, the Factoring process can deal with n sequences simultaneously, so we can deal with permutations of sequences if they are important. We can also make the rules non-deterministic.

3 Algorithm Overview

Before carrying out an empirical analysis of the implemented FI algorithms, we first consider how the algorithms were implemented so the analysis will be understood. Implementation decisions for the general version of the algorithm were based on the following two principles: (1) the performance of the algorithm will be greatly enhanced if we can ensure linear runtime, and (2) the algorithm must be implemented so it can be applied to strings of considerable length without causing problems related to memory consumption.

3.1 Factoring Analysis

The first step in the factoring portion of the algorithm is to read each of the valid symbols in the alphabet from a file, store them in an array that holds all the alphabet symbols, and assign each one a numerical value based upon the index where the symbol was stored. The total number of symbols in the alphabet is b, called the base, due to the fact that alphabet sequences are treated as a numbers represented in the corresponding base number system for the purposes of hashing. In order to implement hashing, a second array of size b^{IB} is created (IB again is the maximum antecedent length defined as inductive base) with each index representing a possible rule antecedent. All indices are initialized to a null value that indicates that the antecedent does not yet have a corresponding consequent. Symbols from the initial string are read one at a time and placed in a queue that maintains the previous symbols read, up to the maximum value of inductive base.

Once the queue fills for the first time each new symbol that is read is treated as the consequent of the antecedent that is implied by the contents of the queue. The sequence of symbols that currently fill the queue are hashed to determine the index of the corresponding antecedent. The current symbol is then compared to the contents of that index; if the index is empty the consequent is placed at the corresponding index. If the index already contains a matching consequent no action is taken. If the index contains a consequent that does not match, the current symbol in the index is given a special value that indicates that the antecedent represented by that index is not a valid antecedent. The indices that are generated by each hash are written to a temporary file each time a hash occurs to serve as input for the next step of the process.

With every possible antecedent having been marked as empty, invalid, or containing a valid consequent, the process of generating the residual for the next level of processing can begin. Each hash index that was previously written to the temporary file is read while simultaneously examining the symbol from the initial string that was being examined when the hash occurred. If the hashed address points to an antecedent index that has been flagged as invalid, the symbol is written to a file as part of the string that will be factored in the next level. With the new string generated for the next level, the remaining task is to output the valid rules based upon the antecedents that have valid consequents in their reduced form. In order to reduce each rule before outputting it, the task of examining every antecedent that could keep the rule from being reduced must be performed (i.e. the rule $BA \rightarrow B$ can be reduced to $A \rightarrow B$ as long as $AA \rightarrow B$, $BA \rightarrow B$, $CA \rightarrow B$, and $DA \rightarrow B$ are all true or do not exist if the input alphabet consists only of A, B, C and D). It is sufficient to state that this can be accomplished by examining the contents of the array that symbolizes all antecedents once for each reduction that is to take place (i.e. a maximum IB of length three would require two passes through this array to find any rules that can be reduced to IB of length one). The entire process described to this point must be repeated for each ruling level that is generated with the exception of loading the alphabet, which occurs once.

3.2 Following Analysis

The following process is far simpler in design than factoring. The alphabet must still be initially loaded, and the array to represent all possible antecedents must still be initialized so that each antecedent is empty. The first ruling level is read from a file and each valid rule that was found is processed so that the consequent is placed in the appropriate index that corresponds to its antecedent. Once the array of antecedents has been filled, the process of reading the target string one symbol at a time, similar to reading the input string in factoring, is performed, in the same manner as the factoring process. The only difference is that an empty antecedent or one that contains a consequent that does not match the current symbol will result in the current symbol being written to a file as a residual for this level. Each new corresponding level requires the repetition of this process with the ruling appropriate for that level and the new target string that was generated by processing the previous target string using the previous level's ruling. Thus when the process is complete the user is left with a file that contains all of the symbols that did not conform to any implicants in the previous level of the ruling.

4 Performance Analysis

With a general understanding of how the factoring and following processes are implemented, we consider the anticipated performance of the algorithm. We are interested in determining if the algorithm as described can be processed in linear time. The size of the input is the primary consideration, and expected runtimes are expressed in terms of input volume.

The factoring process is the more complicated and will necessarily have the longest performance time. The initial pass through the alphabet can be represented in terms of b, the number of valid symbols in the alphabet (also known as the base). The size of the array to represent all possible rule antecedents, namely b^{IB}, must be counted each time the array is examined. This occurs once when initializing each antecedent as empty, and once for each pass to determine if rules can be reduced (IB - 1). Thus we can represent this element of performance by the formulation (IB - 1) (b^{IB}) + b^{IB}. The length of the initial string we will designate by N. The entire length is processed three times for each level: once while scanning symbols to generate consequents, once while examining the hash addresses that were output, and a third time while examining the hash addresses. This requires $2N$ time. Taken together, the algorithm's performance for processing a single level can be described by b + b^{IB} + (IB - 1) (b^{IB}) +

$3N$. We know this process is repeated with each ruling level L that is generated, except when loading the alphabet. This estimate is shown in (6).

$$b + L[b^{IB} + (IB - 1)(b^{IB}) + 3N] \tag{6}$$

For following we need to determine the elements that are factors in its expected performance. The alphabet must still be loaded, and this can again be expressed by the variable b. The antecedent array still exists and can still be represented by b^{IB}; however, the number of times that this array is examined differs. The array is still initialized once while processing the current level, but now it is only examined one additional time as rules are expanded back into their maximum antecedent form. This can be represented simply as $2b^{IB}$. The length of the compared string, N, is only examined once per level in the process of following, but we must define a new variable R that corresponds to the number of rules in the ruling for the level, since each rule must be loaded into the antecedent array. This yields the upper bound on time complexity of $b + R + 2b^{IB} + N$ for each level and (7) for all levels.

$$b + L[R + 2b^{IB} + N] \tag{7}$$

Expressing these equations in terms of Big-O notation yields the equations (8) and (9) for the factoring and following respectively.

$$\mathrm{O}(b + L[b^{IB} + (IB - 1)(b^{IB}) + 3N]) \tag{8}$$

$$\mathrm{O}(b + L[R + 2b^{IB} + N]) \tag{9}$$

We can substitute b^{IB} in for R since this value is the maximum value. Next expanding (8) yields (10), and with the substitution, (9) can be rewritten as (11).

$$\mathrm{O}b + L[IBb^{IB} + 3N]) \tag{10}$$

$$\mathrm{O}(b + L[3b^{IB} + N]) \tag{11}$$

While the term b is variable depending on the problem domain, the value remains constant within any single problem domain (i.e. the algorithm is not designed to apply rulings to strings that are formed from a different alphabet than the string that was examined during the factoring process). Furthermore, the algorithm always utilizes the same maximum IB in the following process as the one used in the factoring process (i.e. the algorithm may be applied to strings of any length, but the maximum IB does not vary between the factoring and following processes). It is therefore possible, based on these facts and the fact the variables are independent of N, to treat these variables as constants. Removing these variables, along with all other constants from both equations, produces the equation (12) for both factoring and following.

$$\mathrm{O}(L + LN) \tag{12}$$

The worst-case scenario of the FI Algorithm can be determined by examining cases where the process of factoring is applied to a string that contains few recognizable patterns or no patterns at all. In the latter case it is evident that the only rule that can possibly be generated is the rule that defines those symbols that start the given string. When this type of string is factored it will create a situation where the number of levels in the ruling will be $\frac{N}{b}$ bounded by N if b is small. Knowing this fact leads to the conclusion that the worst-case scenario of the FI Algorithm is one where the factored string is entirely random, and the expected performance time ($\mathrm{O}(L + LN)$) is quadratic ($\mathrm{O}(N + N^2)$). We now show a strategy to prevent this scenario from occurring.

The expected performance of (12) for the factoring and following portions of the algorithm can be reduced to $\mathrm{O}(N)$ under one of the two following conditions: 1) we can ensure that the term L remains a constant, or 2) we can ensure that L remains an insignificant factor when compared to the variable N. It is possible to begin to satisfy the second condition by restricting the strings being factored by the algorithm to only those that are believed to have significant underlying patterns. However, this is idealistic in the sense that the randomness of the factored string would have to be determined beforehand in order to ensure that this fact remains true. It is a simpler task to allow the user to place an upper bound on the number of levels that can be produced by factoring the given string, thus ensuring that L remains a constant that is equal to or less than this upper bound; it is this strategy that is employed to ensure a linear runtime of the proposed algorithm.

5 Empirical Performance Test

The following subsections discuss the empirical results of the experiment in order to determine the accuracy of the predicted expectations that both processes will perform with linear performance dependent upon the number of input symbols N.

5.1 Experiment Design

The experiment is designed to allow for the testing of whether or not the factoring and following are producing linear runtimes. In order to fulfill these testing requirements, the processes of factoring and following have been executed input data sizes increasing by an order of magnitude (i.e. string lengths of 10^3, 10^4, 10^5, 10^6, 10^7, and 10^8 symbols) using an alphabet that consists of four symbols: A, B, C, and D. In order to determine linear performance regardless of the maximum IB used two IBs (5 and 10). The maximum number of levels that can be processed has been restricted to 100 levels in all cases, and each process is implemented in C++ with the timing mechanism built into the code itself.

5.2 Timing Results

Ten repetitions were done for each experiment with times recorded to the nearest millisecond. Fig. 1 and 2 provide summary results of these experiments for factoring and following respectively. The first task in analyzing the resultant data from the experiment is to determine if the factoring portion of the algorithm is indeed producing slower execution times than the following portion of the algorithm. The graphical data suggests that the factoring process is producing slower times compared to the following process, but the question remains as to how much slower. In analyzing the raw data we obtained from the repeated experiments, we can compare the performance of the factoring portion to the following portion of the algorithm. Our empirical results show that the factoring process required 5.82 times the execution time of the following process when we are dealing with the longer antecedents and a value of 2.49 times for the shorter antecedents.

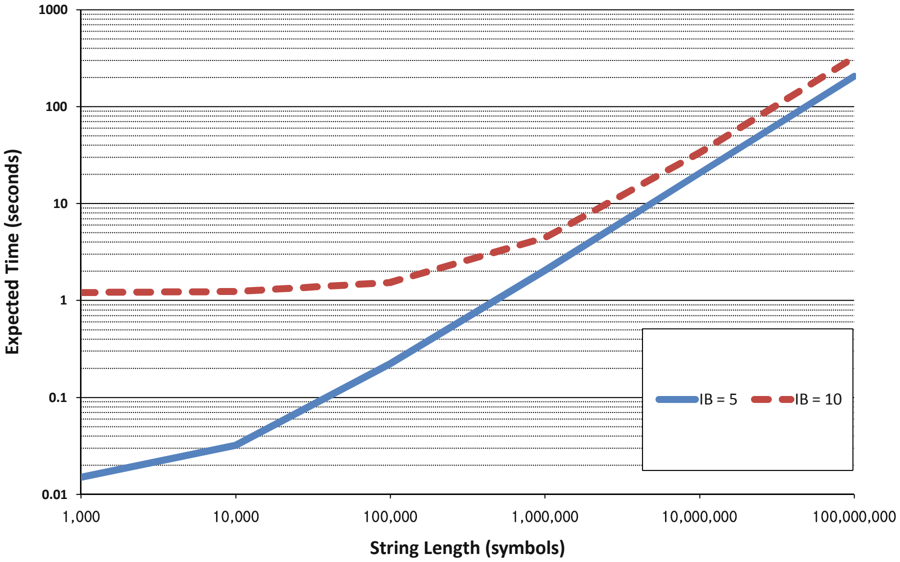


Fig. 1. Data for the Factoring process

The data recorded in both processes that correspond to a maximum IB of length five clearly produced a linear progression (as demonstrated by Fig. 1 and 2), despite the fact that there is an increase in the slope of the line that corresponds to the factoring process once the length of the target string exceeds 10^4 symbols. A linear progression is also reached in both processes using a maximum IB of length ten, but the progression does not become completely linear until the length of the target string has reached 10^6 and 10^5 symbols for factoring and following, respectively.

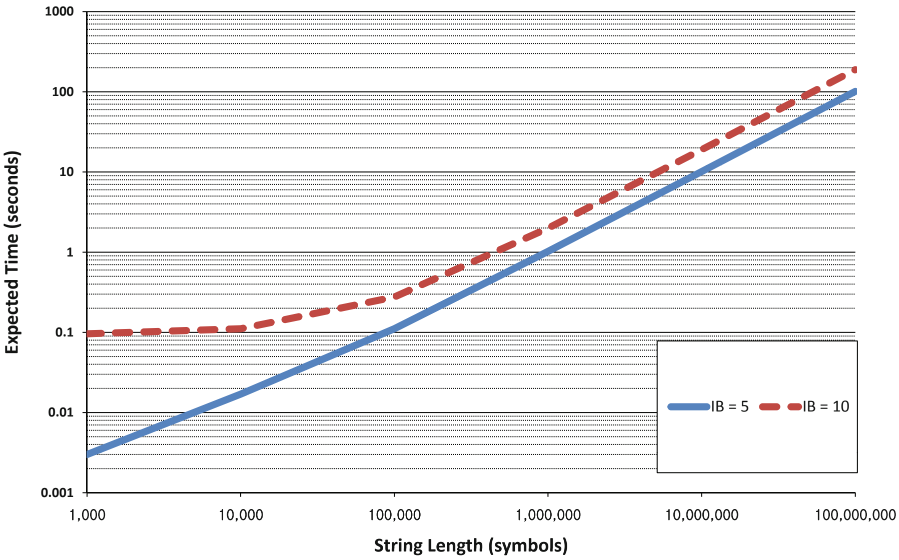


Fig. 2. Data for the Following process

6 Application of FI Algorithm to Actual Data

Consider the nucleotide subsequence obtained from [5] shown in (13).

$$\begin{aligned}&\mathrm{GTGCGATTTTTTTCTCCTCCTTTTTTTACCCTCCCGTT}\\&\mathrm{TTTTTCTTTTCTTTTTTTTTTCCCTATCCTTTTTTTGT}\end{aligned} \quad (13)$$

This subsequence begins at the 244th position in the sequence consisting of some 21,069 symbols. Factoring this sequence we obtained 16,916 rules, meaning that the sequence has 4,153 nucleotides that have duplicate antecedents. This implies 20 % of the subsequences overlap at least with one other subsequence. The ruling built consists of 18 levels, and since this percentage comes from a multi-level ruling, this commonality between subsequences may be due to elimination of symbols at one level producing homogeneous antecedents at the next level. Fig. 3 shows the number of implicants by level. From Fig. 3, we see that the number of implicants stays pretty constant through level 8, and then it grows quickly as the levels increase. This growth can be attributed to the fact that the sequence becomes choppier, that is, the repeated runs of patterns are removed by level 9 and so with more disparity, fewer identical antecedents with differing consequents contradict one another.

From Fig. 3, we see that the number of implicants stays pretty constant through level 8, and then it grows quickly as the levels increase. This growth can be attributed to the fact that the sequence becomes choppier, that is, the repeated runs of patterns are removed by level 9 and so with more disparity, fewer identical antecedents with differing consequents contradict one another. For the

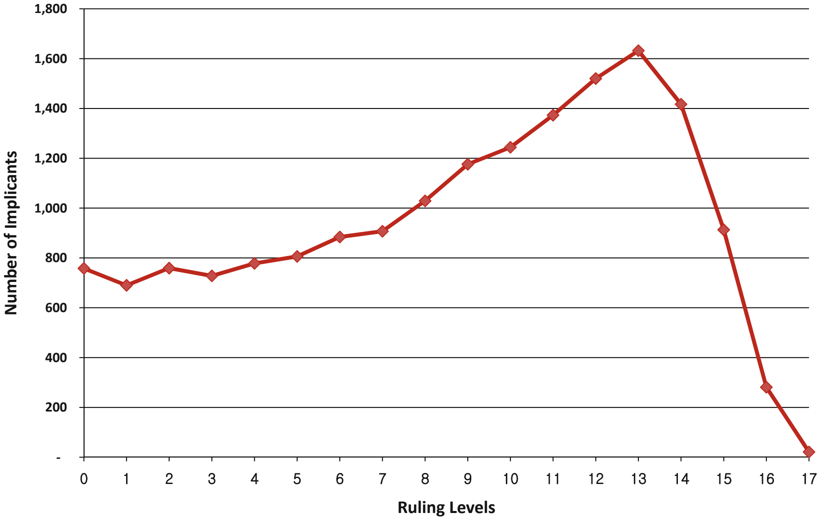
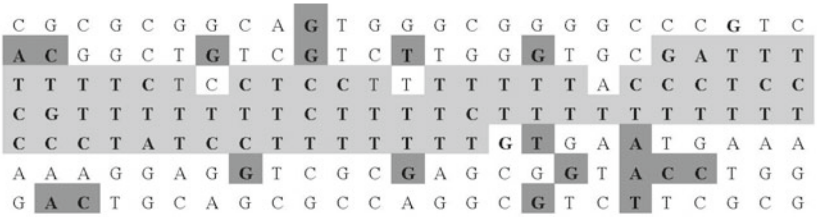


Fig. 3. Number of implicants per each level in the ruling

next step, we factored the subsequence in (13) and used this as the ruling with an inductive base of length 3, and then processed the entire sequence. Fig. 4 provides the results of this activity. The light gray cells are those that belong to (13), and the dark gray are those symbols that have an implicant matching one in the ruling but are not in the subsequence of interest. The basis for this type of application is well treated in [6].



C G C G C G G C A G T G G G C G G G G C C C G T C
A C G G C T G T C G T C T T G G G T G C G A T T T
T T T T C T C C T C C T T T T T T T A C C C T C C
C G T T T T T T T C T T T T C T T T T T T T T T T
C C C T A T C C T T T T T T T G T G A A T G A A A
A A A G G A G G T C G C G A G C G G T A C C T G G
G A C T G C A G C G C C A G G C G T C T T C G C G

Fig. 4. Nucleotide sequence starting at position 201 and ending at position 375

Besides the subsequence in Fig. 4, there are other matches. Fig. 5 shows two of the longer ones. The first subsequence in Fig. 5 starts at position 54 and the second starts at position 935. Since the identification of a matching substring is not difficult, we provide an extension to the matching under random permutations of the subsequence being used to build the search ruling. We modified 10 % of the symbols in the substring we were looking for and then followed the unknown string with a ruling of inductive base 3 and another with inductive base 9. We

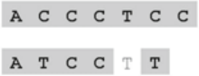


Fig. 5. Two subsequences found by the ruling with white indicating no match

provide the results in Fig. 6, where we only show that portion associated with the location of the substring we are trying to find. In Fig. 7 we show another contiguous substring from the unknown string, so that the density of the two areas can be compared. Comparing the results from these two test blocks, we obtain the results shown in Table 1. We have also compared the complexity of such nucleotide sequences by considering their representation within a ruling as a measure not unlike Kolmogorov complexity [7].

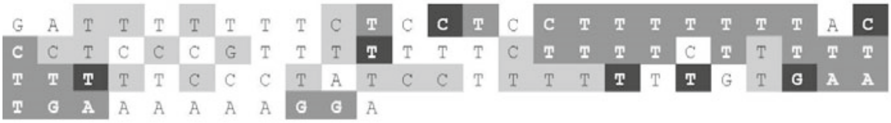

G A T T T T T T T C T C C T C C T T T T T T T A C
C C T C C C G T T T T T T T C T T T T C T T T T T
T T T T T C C C T A T C C T T T T T T T G T G A A
T G A A A A A A G G A

Fig. 6. Results of Following when 10 % of the symbols are changed. White indicates no change, dark gray are symbols recognized by ruling of IB of 9, light gray are recognized symbols by ruling of IB 3, and 50 % gray are symbols recognized by both rulings.

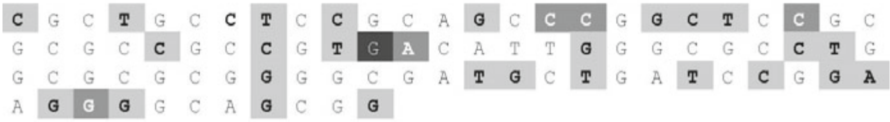

C G C T G C C T C C G C A G C C C G G C T C C G C
G C G C C G C C G T G A C A T T G G G C G C C T G
G C G C G C G G G G C G A T G C T G A T C C G G A
A G G G G C A G C G G

Fig. 7. Results starting from position 450 in the unknown sequence where the color key is identical to that of Fig. 6

Table 1. Comparative counts for the data of Fig. 6 and 7

	IB	IB		
	3	9	Both	White
Fig. 6	22	7	26	30
Fig. 7	26	1	4	54

7 Conclusion

The results of this work have been to provide a comparative basis for the timing of an algorithm that will recognize substrings of symbols, even under mutation. We have shown by logical argument as well as by empirical data that the algorithm operates in linear time with the size of the input data sequence being the driving factor. Also as shown in the last section in Table 1, the short inductive base provides too many extraneous symbol matches, while the long inductive base provides too few. There must be another inductive base that would be most appropriate. But even with these two selections, and the selections were made to first yield a ruling with only one level (IB = 9) and as many levels as we could obtain (IB = 3), the results provide an upper and lower bound. We still have more work to do to refine this algorithm to provide a more robust result for processing large sequences of symbols from a small alphabet. It is clear that the longer the sequence, the more potential there is for conflict when the inductive base is fixed to a reasonable value, perhaps 5 to 7 symbols. We have shown that such conflicts indeed do exist in the early levels of a ruling, limiting their growth. Lastly, we believe that the approach of non-deterministic rulings may provide an additional benefit for this kind of processing.

References

1. Buhler, J., Tompa, M.: Finding Motifs Using Random Projections. Journal of Computational Biology 9(2), 225–242 (2002)
2. Li, G., Liu, B., Xu, B.: Accurate Recognition of CIS-regulatory motifs with the Correct Lengths in Prokaryotic Genomes. Nucleic Acids Research 38(2), e12 (2009), http://nar.oxfordjournals.org
3. Fisher, P.S., Fisher, H., Baek, J., Angaye, C.: Syntactic Pattern Recognition Using Finite Inductive Strings. In: Kadirkamanathan, V., Sanguinetti, G., Girolami, M., Niranjan, M., Noirel, J. (eds.) PRIB 2009. LNCS (LNBI), vol. 5780, pp. 89–101. Springer, Heidelberg (2009)
4. Case, J.H., Fisher, P.S.: Long Term Memory Modules. Bulletin of Mathematical Biology 46(2), 295–326 (1984)
5. Source used for data in this paper, http://www.ncbi.nlm.nih.gov/nuccore/12751174
6. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. Journal of Molecular Biology 147, 195–197 (1981)
7. Fisher, P.S., Baek, J., Adeyeye, J., Setubal, J.: Finite Inductive Sequences, Kolmogorov Complexity with Application to Genome Sequences. In: International Conference on Bioinformatics, Computational Biology, Genomics and Chemoinformatics, BCBGC-10 (July 2010)

An Algorithm to Find All Identical Motifs in Multiple Biological Sequences

Ashish Kishor Bindal[1], R. Sabarinathan[1], J. Sridhar[2],
D. Sherlin[1], and K. Sekar[1,⋆]

[1] Bioinformatics Centre (Centre of excellence in Structural Biology and Bio-computing), Indian Institute of Science, Bangalore 560012, India
Tel.: +91-080-22933059/23601409; Fax: +91-080-23600683/23600551
[2] Center of Excellence in Bioinformatics, School of Biotechnology, Madurai Kamaraj University, Madurai 625021, Tamilnadu, India
sekar@physics.iisc.ernet.in, akbindal@ug.iiita.ac.in,
sabari.binc@gmail.com, srimicro2002@gmail.com, sherlinsugirtha@gmail.com,
http://www.physics.iisc.ernet.in/~dichome/sekhome/index.html

Abstract. Sequence motifs are of greater biological importance in nucleotide and protein sequences. The conserved occurrence of identical motifs represents the functional significance and helps to classify the biological sequences. In this paper, a new algorithm is proposed to find all identical motifs in multiple nucleotide or protein sequences. The proposed algorithm uses the concept of dynamic programming. The application of this algorithm includes the identification of (a) conserved identical sequence motifs and (b) identical or direct repeat sequence motifs across multiple biological sequences (nucleotide or protein sequences). Further, the proposed algorithm facilitates the analysis of comparative internal sequence repeats for the evolutionary studies which helps to derive the phylogenetic relationships from the distribution of repeats.

Keywords: Sequence motifs, nucleotide and protein sequences, identical motifs, dynamic programming, direct repeat and phylogenetic relationships.

1 Introduction

A conserved pattern of a nucleotide or amino acid sequence with a specific biological function is known as a sequence motif and is becoming increasingly important in the analysis of gene regulations [1]. Research on protein and DNA sequences revealed that specific sequence motifs in biological sequences exhibit important characteristics [2]. In DNA sequences, the sequence motif act as specific binding sites for proteins (nuclease, transcription factors, etc.) and RNAs (mRNA splicing, transcription termination, etc.) [1]. Further in proteins, these motifs act as enzyme catalytic sites, prosthetic group attachment sites (haem, pyridoxal phosphate, biotin, etc.), metal binding amino acids, cysteines involved

[⋆] Corresponding author.

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 137–148, 2010.

in disulfide bonds or regions involved in binding a molecule [3]. In the recent years, due to the exponential rise in the volume of nucleotide and amino acid sequences in their respective databases, identification of sequence motifs using experimental methods is impossible. In addition, many newly discovered protein sequences do not share a global sequence similarity with a known protein. However, they share a short stretch of conserved sequences which represent the characteristics of similar domains [4]. Over the past years, these problems have been addressed using newly developed computational methods [5],[6],[7]. To this end, an efficient algorithm is proposed using the dynamic programming.

Earlier studies indicate that the transcription factor (TF) binding sites are well conserved motifs of short DNA sequence stretch. The motif size ranges from 5 to 35 nucleotides long and occur in a well-ordered and regularly spaced manner [8],[9]. For example, in eukaryotes the cis-regulatory module (CRMs) usually occurs in a fixed arrangement and distributed over very large distances. Further, the repeat occurrence of this binding site will help for the alternate modes of binding by the same protein which leads to the regulation of transcriptional activity. Gene duplications and recombination events are thought to be responsible for this repeat occurrence of sequence motifs. The distribution of repeats in archaea indicates that they have an intermediate relationship between prokaryotes and eukaryotes [10]. In DNA, these repeats are mainly classified into two groups such as tandem and interspersed repeats. The tandem repeats are an array of consecutive repeats and often associated with disease syndromes [11]. On the other hand, interspersed repeats are copies of transposable elements located at various regions in a genome. Moreover, the repeats that are separated by intermediate sequences of constant length occurring in clusters are referred to short regularly spaced repeats (SPSRs) [12]. Generally, these short repeats indicate the position of deletion and precise removal of transposable elements [13], where as, longer identical repeats are responsible for class switching in immunoglobulins [14]. Further, tandem repeats in telomers are involved in the protection of chromosome end and its length. In some cases, the internal sequence repeats in proteins adopt similar three-dimensional structures [15],[16]. However, further work is necessary to ascertain this aspect. In addition, the internal sequence repeats are observed to be associated with structural motifs or domains in the class of repeat protein families [17]. Further, the repeated sequence motifs play an active role in protein and nucleotide stability, thus, not only ensuring proper functioning [18] but some times cause malfunction and disease [19],[20].

1.1 Existing Algorithms

In the post genomic era, many algorithms are available in the literature to find the sequence motifs and repeats in biological sequences. However, these algorithms significantly vary in their methodologies. In general, the motif finding algorithms are divided into two major groups based on their working principle. The first group of algorithms identifies the motifs with reference to the annotated motif database. For example, the programs InterProScan [5], Motif Scan [21], ScanProsite [22] and SMART [23] search for motifs against protein profile database such as Prosite, Pfam, TMHMM etc. In addition, the above mentioned programs are limited to only protein sequences. Further, the program MOTIF [24] identifies the motif in both protein and DNA sequences using the above profile databases as well as user defined libraries. In contrast, another set of programs such as MEME [6], TEIRESIAS [7], ALIGN ACE [25], DILIMOT [26] and Gibbs Sampler [27] identify the motifs without any reference database. However, they use some statistical methods to identify the motifs and represent the conserved regions of the motifs in the form of sequence patterns using regular expressions or sequence logos. It is to note that most of these algorithms lack in the limitation of input sequence size (TEIRESIAS and ALIGN ACE take around 3,50,000 residues and the program MEME limits only to 60,000 residues).

The proposed algorithm has been developed by keeping the above lacuna in mind and uses the dynamic programming method implemented earlier [28] to identify all identical motifs present in multiple biological sequences (nucleotides and protein). To the best knowledge of authors, there is no such algorithm exists in the open literature. The proposed algorithm can be effectively used for the comparative identification of direct repeat motifs in several biological sequences. However, inorder to reduce the computational time, the total number of residues for a single run is restricted to a maximum of 10,00,000 residues.

2 Methodology

The proposed algorithm identifies all motifs which are present in a given set of biological sequences. Since, the problem of finding identical motifs in multiple sequences is similar to the problem of finding identical internal repeats in a sequence, when all sequences are concatenated with a delimiter or special character (z), where z $\notin \sum$ ($\sum$ represents a set of alphabet characters in the input sequences). The criteria for the identical motif should be an exact pattern repeated more than one sequence. Thus, we will refer the identical motif as identical repeat in the following sections. The algorithm adopts the methodology of FAIR algorithm [28]. In addition, it has been improved by using hash table to reduce the time complexity. The working principle of the new methodology is explained in the subsequent sections.

2.1 Pre-processing Phase

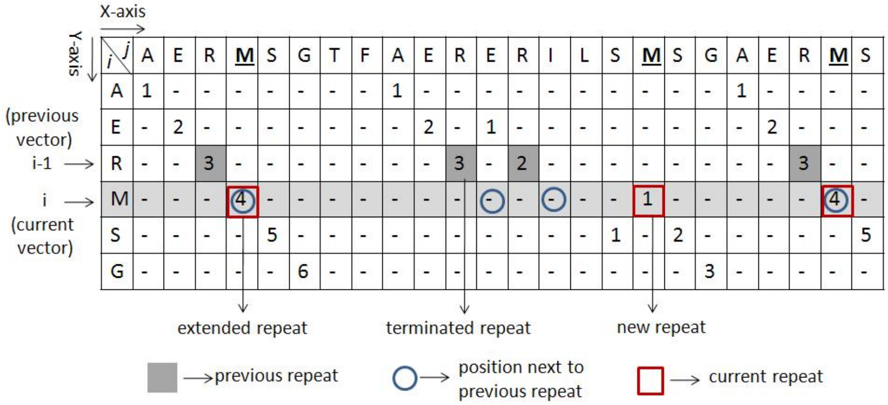
Initially, the uploaded sequences are concatenated with a delimiter at the end of each sequence and stored in a string S. In addition, the starting position of each input sequence in the string S is stored in an array. Further, a hash table is created to improve the execution time during search phase and to store the positions or occurrences of each alphabet (X) in the string S. The size of the hash table is equal to the length of the string S. The number of entries in the hash table varies for DNA (only four A,T,G and C) and protein (20 amino acids) sequences. All the positions of a single character (X) present in the string S are stored in a hash element or key (hash[X]). For example, the hash[X] represents the hash key of character X, the vector Voccurence[hash[X]] contains the occurences or positions of character X in string S and referred as Voccurence[X].

2.2 Searching Phase

The proposed algorithm uses the dynamic programming method to determine the identical repeats in the string S. The string S is aligned itself by taking the same on both X- and Y-axes in a two-dimensional space (see Fig. 1). Instead of creating a two-dimensional matrix for storing the match score values, the algorithm uses the concept of linear space complexity deployed in FAIR algorithm [28] by using two vectors (current and previous). The size of the current and previous vectors is equal to N (length of the string S). While scanning, each element (S[i]) in the Y-axis is used as a probe to search for the match along X-axis (S[j]), where i,j $\in$ 0 $\leq$i$\leq$N, 0$\leq$j$\leq$N (see Fig. 1). During this process, when an element S[i] from Y-axis is matched identically with the element S[j] of X-axis, a hit value of one is added to the value of $j-1^{th}$ in the previous vector and the total is assigned to the j^{th} position of current vector. Thus, the current vector holds the present repeat length with respect to the character S[i] and the previous vector holds the repeat diagonal up to S[i-1]. Whenever a match is not found or the sequence ends (j==N), the value of the previous vector is checked for the size greater than the minimum length of the motif, then the previous vector value is stored as the length of the repeat (L) and the positions of i-1 and j-1 are stored as repeat end positions ($R_{i-1,j-1,L}$). The above operation is repeated recursively till the end of i along Y-axis. The pseudo-code for the recursive operation is given below,

```
IF S[i] equals to S[j] THEN
    set current[j] to previous[j-1]+1;
ENDIF
ELSE
 IF previous[j-1]>=minimum of motif length
    set repeat length (L) to previous[j-1];
    set first repeat end to i-1;
    set second repeat end to j-1;
 ENDIF
END ELSE
```

Advantage of using hash table: Since the current and previous vectors are sparse, the recursive operation at each i (along Y-axis) and j (along X-axis) takes more time for longer sequence. In order to optimize the execution time, a new methodology has been implemented for scanning phase using hash table. The above recursive operation is carried out for each i against X-axis and is only for some j's which are the positions of character (S[i]) in string S. i.e., Voccurences[S[i]] (see Fig. 1). It is explained by using the following lemma: Lemma 1 states: for each i (0$\leq$i$\leq$N), the algorithm checks only the positions next to the previous repeat (see Fig. 1) and at all positions of character S[i], instead for all j (0≤i≤N). As a proof, there can be only three possibilities at each i, such as: (A) any previous repeat can be continued or extended (extended repeat), (B) previous repeat can be terminated and are needed for output (terminated repeat) and finally (C) any new repeat can be a start (see Fig. 1). The above three repeat possibilities are further classified in two sections: (a) for possibility A and C, the match of S[i] and S[j] need to be identical. Further, the current vector of j^{th} element attains the length L > 0, represents the current repeat ($R_{i,j,L}$). Thus, the current repeat is a start position of a repeat or in the part of a continuous or extended repeat. (b) In case of B, termination of previous repeat ($R_{i-1,j-1,L}$), S[i] and S[j] does not match or the length of j equal to the string S. It means that the next position to the previous repeat does not match with the positions of S[i] in (Voccurence[S[i]]) or the repeat is terminated at the end of the sequence and are referred as terminated repeats ($R_{i-1,j-1,L}$). Thus, the



j / i	A	E	R	**M**	S	G	T	F	A	E	R	E	R	I	L	S	**M**	S	G	A	E	R	**M**	S
A	1	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-
E	-	2	-	-	-	-	-	-	-	2	-	1	-	-	-	-	-	-	-	-	2	-	-	-
R	-	-	3	-	-	-	-	-	-	-	3	-	2	-	-	-	-	-	-	-	-	3	-	-
M	-	-	-	4	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	4	-
S	-	-	-	-	5	-	-	-	-	-	-	-	-	-	-	1	-	2	-	-	-	-	-	5
G	-	-	-	-	-	6	-	-	-	-	-	-	-	-	-	-	-	-	3	-	-	-	-	-

Fig. 1. A sample sequence is aligned along X- and Y- axes in a two-dimensional space. The number of repeat possible (current, previous and terminated) is highlighted.

algorithm scans only in the regions of positions next to previous repeat and all positions of character S[i]. The following steps are carried for each iteration of i with respect to lemma 1;

1. updation of current vector due to current repeat from (a) of lemma 1. current[j]=previous[j-1]+1; ∀ j ∈ Voccurrence[S[i]]
2. finding terminating repeat from (b) of lemma 1. if(previous[j-1] > minimum length of repeat AND j ∉ Voccurrence[S[i]]) then Vterminated.push(j-1); ∀ j-1 ∈ Voccurrence S[i-1]]

The Vterminated vector stores all the end positions of terminated repeats. Moreover, the algorithm performs the above two operations together by merging Voccurence[S[i]] and Voccurence[S[i-1]].

Pseudo code (entire searching phase using hash table)

```
WHILE (m < Voccurrence[S[i]].size() && n < Voccurrence[S[i-1]].size() )
 IF Voccurrence [S[i]][m]] == Voccurence [S[i-1]][n]]+1) THEN
 //current repeat
 set current[Voccurrence[S[i]][m]] to previous[Voccurrence [S[i-1][n]]+1;
 set m to m + 1; set n to n + 1;
 ENDIF
 ELSE IF Voccurence[S[i]][m]] < Voccurence[S[i-1]][n]+1] THEN
 //current repeat with no previous repeat found
 current[Voccurence[S[i][m]]=1;
 Set m to m+1;
 END ELSE IF
 ELSE IF Voccurence[S[i][m]] < Voccurence[S[i-1]][n]]+1 THEN
 // no current repeat found and this is terminating repeat
 IF previous[Voccurence[S[i-1][n]] >= minimum length of repeat THEN
 Vterminated.push(Voccurence[S[i-1][n]]);
 ENDIF
 Set n to n+1;
 END ELSEIF
 ENDWHILE
```

2.3 Post Processing Phase

In this section, for each terminated repeat $(R_{i-1,j-1,L})$ in Vterminated vector, the repeat length (L) is checked against the length of all the repeats in a previous vector. If the length is greater than or equal to L (terminated repeat length), then all such previous repeat $(R_{i-1,j'-1,L})$ positions are stored in a data structure motif. These motif are pushed into a vector Vmotif. Further, the value of the vector Vmotif is sorted (on the basis of repeat string) using a built-in STL (Standard Template Library) function. Finally, unique motifs are determined after removing all the redundant entries. The detailed output of the algorithm contains the length of the motifs and their start and end positions.

2.4 Time Complexity

The computational or time complexity of the algorithm is explained below based on the following; **Preprocessing:** In this phase, the positions of each alphabets in the string S is identified to create a hash table and the scanning process is performed in one-dimensional space with O(N) time complexity, where N is the length of the string S. **Scanning:** As explained earlier, the algorithm performs the scanning operations together by merging Voccurence[S[i]] and Voccurence[S[i-1]]. Thus for each i, the scanning sequence along X-axis takes $O(2N/|\sum|)$ time and $\sum$ represents the alphabets in string S, where $|\sum|$ represents the size of $\sum$ set. During each iteration along Y-axis, the value of current vector is assigned into previous vector and the current vector is reinitialized to zero, results in $O(N/|\sum|)$ time complexity. In addition, the process of Vterminated repeats requires scanning along X-axis results again in $O(N/|\sum|)$ complexity. Thus, the entire scanning phase takes $O(4N/|\sum|)$ time to find identical repeats in the string S. **Post processing:** In this section, the motif stored in Vmotif is sorted using the STL sort function which results in (N'logN') time complexity (where N' is number of repeats). However, the execution time of STL sort is less compared to that of the above steps. Considering the above three cases, the algorithm follows $O(4N^2/|\sum|)$ time complexity to find the identical repeats using hash table. The algorithm is more effective with an increase in $|\sum|$ and is improved over the existing algorithm, FAIR [28].

3 Results and Discussion

3.1 Case Study 1

To test the efficiency of the proposed algorithm, a set of eight major CRISPR (Clustered Regularly Interspaced Short Palindromic Repeats) nucleotide sequences is considered in this case study. The CRISPRs are direct repeats (identical repeats) with a length ranges from 24 to 48 nucleotides and the repeats in DNA are separated by spacers of similar length. These repeats are commonly present in many bacteria and archaea groups which help for the acquired resistance against phages [29]. The CRISPR sequences used in the present study are taken from four different species such as *Salmonella typhimurium* LT2*, Salmonella enteric serovar Typhi* Ty2*, Salmonella enteric serovar Paratyphi* A Str. AKU_12601 and *Salmonella enteric Choleraesuis*. The sequences are of various lengths with a minimum and maximum of 212 and 1982 nucleotides respectively. The input parameters provided for the search are: (a) the length of motif to be searched (for example: greater than or equal to 30) and (b) the minimum number of motif multiplicity (for example: greater than or equal to two). The motif multiplicity is defined as the number of times a motif is repeated in all the given sequences. The proposed algorithm identified 118 possible motifs in all four CRISPR sequences from four different species (*Salmonella typhimurium*LT2 *, Salmonella enteric serovar Typhi* Ty2, *Salmonella enteric serovar Paratyphi* A Str. AKU_12601 and *Salmonella enteric Choleraesuis*). A sample output (only a part of the output is shown for clarity) of the result is shown below.

```
----------------------------------------------------------------------
Input file name: fasta.txt
Length of motif: greater than 30
Motif multiplicity: greater than 2
Output file name: out.txt
----------------------------------------------------------------------
Motif: AACGGTTTATCCCCGCTGGCGCGGGGAACAC
Motif length: 31
Motif Occurrences :7
Present in 3 Sequences
>CRISPR-1, SALMONELLA TYPHIMURIUM LT2
Position(s): [304,334]
>CRISPR-2, SALMONELLA TYPHIMURIUM LT2
```

```
Position(s): [427,457] [549,579]
>CRISPR-3, SALMONELLA TYPHIMURIUM LT2
Position(s): [243,273] [793,823] [1586,1616] [1891,1921]
----------------------------------------------------------------------
Motif : CGGTTTATCCCCGCTGGCGCGGGGAACACA
Motif length :30
Motif Occurrences:15
Present in 6 Sequences
>CRISPR-1, SALMONELLA TYPHIMURIUM LT2
Position(s): [306,335]
>CRISPR-2, SALMONELLA TYPHIMURIUM LT2
Position(s): [673,702]
>CRISPR-3, SALMONELLA TYPHIMURIUM LT2
Position(s): [612,641] [856,885] [1039,1068] [1100,1129]
             [1405,1434] [1466,1495]
>CRISPR-2, SALMONELLA CHOLERAESUIS
Position(s): [306,335]
>CRISPR-1, SALMONELLA PARATYPHI A
Position(s): [184,213] [306,335] [367,396]
>CRISPR-1 SALMONELLA TYPHI TY2
Position(s): [62,91] [184,213] [245,274]
----------------------------------------------------------------------
Motif : GCGGTTTATCCCCGCTGGCGCGGGGAACAC
Motif length :30
Motif Occurrences :23
Present in 5 Sequences
>CRISPR-2, SALMONELLA TYPHIMURIUM LT2
Position(s): [123,152] [489,518] [611,640] [672,701] [733,762]
             [795,824] [856,885]
>CRISPR-3, SALMONELLA TYPHIMURIUM LT2
Position(s): [61,90] [183,212] [611,640] [733,762] [1038,1067]
             [1099,1128] [1221,1250] [1343,1372] [1709,1738]
>CRISPR-2, SALMONELLA CHOLERAESUIS
Position(s): [244,273]
>CRISPR-1, SALMONELLA PARATYPHI A
Position(s): [122,151] [183,212] [244,273] [427,456]
>CRISPR-1 SALMONELLA TYPHI TY2
Position(s): [122,151] [244,273]
----------------------------------------------------------------------
```

It is interesting to note that, the above motif of length 30 residues, GCGGTTTATCCCCGCTGGCGCGGGGAACAC, clearly shows the efficiency of the proposed algorithm in finding the motif in all possible locations of the chosen four nucleotide sequences. Firstly, the different motif locations identified in the sequences of CRISPR-2 of *Salmonella typhimurium* LT2 and CRISPR-1 of *Salmnoella Paratyphi* A are found to be separated by an approximate spacer of length 32 nucleotides. Further, it is also to note that the occurrence of the motif is nearly conserved at the same locations (123 to 152) and (244 to 273). However, the number of motifs in each sequence varies (minimum = 1 and maximum 9) and represent genome variations among the four species.

3.2 Case Study 2

A total of three hexokinase-1 protein sequences from orthologous species such as *Homo Sapiens, Mus Musculus* and *Rattus norvegicus* are considered in this case study. The minimum length of motif to be searched is given as greater than or equal to 5 and the motif multiplicity is given as two (by default). The proposed algorithm identified 85 identical motifs (only part of the output is shown below) present in all the three sequences. Interestingly, a total of 17 out of 85 identified motifs are repeated more than once in the same protein sequence (see below for details). A sample output of the repeat motifs (17) is shown below.

```
---------------------------------------------------------------------
Input file name: fasta.txt
Length of motif: greater than 5
Motif multiplicity: greater than 2
Output file name: out.txt
---------------------------------------------------------------------
Motif : FVRSIPDG
Motif length :8
Motif Occurences:4
Present in 3 Sequences
>gi|188497754|REF|NP_000179.2|[HOMO SAPIENS]
Position(s) : [67,74]
>gi|148700161|GB|EDL32108.1| [MUS MUSCULUS]
Position(s) : [66,73]
>gi|6981022|REF|NP_036866.1| [RATTUS NORVEGICUS]
Position(s) : [67,74] [515,522]
---------------------------------------------------------------------
Motif : GSGKGAA
Motif length :7
Motif Occurrences:5
Present in 3 Sequences
>gi|188497754|REF|NP_000179.2|[HOMO SAPIENS]
Position(s) : [448,454] [896,902]
>gi|148700161|GB|EDL32108.1| [MUS MUSCULUS]
Position(s) : [447,453] [895,901]
>gi|6981022|REF|NP_036866.1| [RATTUS NORVEGICUS]
Position(s) : [896,902]
---------------------------------------------------------------------
Motif : GFTFSFPC
Motif length :8
Motif Occurrences :6
Present in 3 Sequences
>gi|188497754|REF|NP_000179.2|[HOMO SAPIENS]
Position(s) : [151,158] [599,606]
>gi|148700161|GB|EDL32108.1| [MUS MUSCULUS]
Position(s) : [150,157] [598,605]
```

```
>gi|6981022|REF|NP_036866.1| [RATTUS NORVEGICUS]
Position(s) : [151,158] [599,606]
------------------------------------------------------------------------
Motif : VAVVNDTVGTMMTC
Motif length :14
Motif Occurrences :6
Present in 3 Sequences
>gi|188497754|REF|NP_000179.2|[HOMO SAPIENS]
Position(s) : [204,217] [652,665]
>gi|148700161|GB|EDL32108.1| [MUS MUSCULUS]
Position(s) : [203,216] [651,664]
>gi|6981022|REF|NP_036866.1| [RATTUS NORVEGICUS]
Position(s) : [204,217] [652,665]
------------------------------------------------------------------------
```

The above results clearly show that the repeat motifs are conserved in all three sequences used. It is interesting to note, the three-dimensional structure of the last two motifs (GFTFSFPC and VAVVNDTVGTMMTC) repeated twice in *Homo Sapiens* and are superposed well with a root mean square deviation of $0.17\AA$ and $0.27\AA$ [16]. Further, the above results have been compared with the results of sequence alignment programs such as BLASTP [30] and CLUSTALW [31]. The output (results not shown) of these programs shows that the orthologous sequences exhibit high sequence similarity of more than 95%. Thus, the sequences are aligned end to end which leads to complexity in identifying the repeated motifs.

4 Implementation

The algorithm requires three inputs: a file of nucleotide or protein sequences in FASTA format, the length of the sequence motif to be searched and the number of motif multiplicity. The proposed algorithm generates a detailed output containing the location of motifs in each sequence. An option is also provided for the users to remove the redundant entries from the given input sequences. For example, only one sequence will be considered if two of the given or uploaded input sequences are having sequence identity of more than or equal to 90%. Due to less time complexity of proposed algorithm, there is no limitation in the number of motifs to be identified. The proposed algorithm has been written in C++ and successfully tested on a Linux box (Fedora core 9 and Red hat 9.0) and Solaris (10.0) environments. A standalone version of the proposed algorithm can be obtained upon request by sending an E-mail to the corresponding author Dr. K. Sekar (sekar@physics.iisc.ernet.in). In the future, we also plan to create an internet computing server for the proposed algorithm.

5 Conclusion

The algorithm finds the identical motifs in both nucleotide and proteins sequences. It has been developed with a broad view in mind to provide a comprehensive solution to the task of finding conserved as well as direct repeat motifs in a given multiple biological sequences. Further, the algorithm helps to analyze the differences in repeat numbers in various genomes and provides an insight to the horizontal gene transfer events during microbial evolution. One of the potential applications of this work is the comparative study of transposons in different sub species which provides a trace for the analysis of gene duplication.

Acknowledgements

The authors gratefully acknowledge the use of the Bioinformatics Centre (DIC) and the Interactive Graphics Based Molecular Modeling facility (IGBMM). These facilities are supported by the Department of Biotechnology, Government of India, India. One of the authors (KS) thank the Department of Information Technology (DIT) for funding in the form of a research grant.

References

1. D'Haeseleer, P.: What are DNA sequence motifs? Nat. Biotechnol. 24, 423–425 (2006)
2. Kumar, C., Kumar, N., Sarani, R., Balakrishnan, N., Sekar, K.: A Method to find Sequentially Separated Motifs in Biological Sequences (SSMBS). In: Chetty, M., Ngom, A., Ahmad, S. (eds.) PRIB 2008. LNCS (LNBI), vol. 5265, pp. 13–37. Springer, Heidelberg (2008)
3. Hulo, N., Sigrist, C.J., Le Saux, V., Langendijk-Genevaux, P.S., Bordoli, L., Gattiker, A., De Castro, E., Bucher, P., Bairoch, A.: Recent improvements to the PROSITE database. Nucl. Acids Res. 32, D134–D137 (2004)
4. Huang, J.Y., Brutlag, D.L.: The EMOTIF database. Nucl. Acids Res. 29, 202–204 (2001)
5. Zdobnov, E.M., Apweiler, R.: InterProScan–an integration platform for the signature-recognition methods in InterPro. Bioinformatics 17, 847–848 (2001)
6. Bailey, T.L., Elkan, C.: Fitting a mixture model by expectation maximization to discover motifs in biopolymers. Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology 2, 28–36 (1994)
7. Rigoutsos, I., Floratos, A.: Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm. Bioinformatics 14, 55–67 (1998)
8. Werner, T.: Model for prediction and recognition of eukaryotic promoters. Mamm. Genome 10, 168–175 (1999)
9. VanHelden, J., Andre, B., Collado-Vides, J.: Extracting Regulatory Sites from the Upstream Region of Yeast Genes by Computational Analysis of Oligonucleotide Frequencies. J. Mol. Biol. 281, 827–842 (1998)
10. Koonin, E.V., Mushegian, A.R., Galperin, M.Y., Walker, D.R.: Comparison of archeal and bacterial genomes: Computer analysis of protein sequence predicts novel function and suggests chimeric origins for the archaea. Mol. Microbiol. 25, 619–637 (1997)
11. Boby, T., Patch, A.M., Aves, S.J.: TRbase: a database relating tandem repeats to disease genes in the human genome. Bioinformatics 21, 811–816 (2005)
12. Mojica, F.J., Diez-Villasenor, C., Soria, E., Juez, G.: Biological significance of a family of regularly spaced repeats in the genomes of archaea, bacteria and mitochondria. Mol. Microbiol. 36, 244–246 (2000)
13. Van de Lagemaat, L.N., Gagnier, L., Medstrand, P., Mager, D.L.: Genomic deletions and precise removal of transposable elements mediated by short identical DNA segments in primates. Genome Res. 15, 1243–1249 (2005)

14. Wu, T.T., Miller, M.R., Perry, H.M., Kabat, E.A.: Long identical repeats in the mouse gamma 2b switch region and their implications for the mechanism of class switching. EMBO J. 3, 2033–2040 (1984)
15. Banerjee, N., Chidambarathanu, N., Sabarinathan, R., Michael, D., Vasuki Ranjani, C., Balakrishnan, N., Sekar, K.: An Algorithm to Find Similar Internal Sequence Repeats. Curr. Sci. 97, 1345–1349 (2009)
16. Sarani, R., Udayaprakash, N.A., Subashini, R., Mridula, P., Yamane, T., Sekar, K.: Large cryptic internal sequence repeats in protein structures from Homo sapiens. J. Biosciences 34, 103–112 (2009)
17. Sabarinathan, R., Basu, R., Sekar, K.: ProSTRIP: A method to find similar structural repeats in three-dimensional protein structures. Comput. Biol. Chem. 34, 126–130 (2010)
18. Heringa, J.: Detection of internal repeats: How common are they? Curr. Opin. Struct. Biol. 8, 338–345 (1998)
19. Djian, P.: Evolution of simple repeats in DNA and their relation to human diseases. Cell 94, 155–160 (1998)
20. Pons, T., Gomez, R., Chinea, G., Valencia, A.: Beta-propellers: associated functions and their role in human diseases. Curr. Med. Chem. 10, 505–524 (2003)
21. MOTIF SCAN, http://myhits.isb-sib.ch/cgi-bin/motif_scan
22. de Castro, E., Sigrist, C.J., Gattiker, A., Bulliard, V., Langendijk-Genevaux, P.S., Gasteiger, E., Bairoch, A., Hulo, N.: ScanProsite: detection of PROSITE signature matches and ProRule-associated functional and structural residues in proteins. Nucl. Acids Res. 34, W362–W365 (2006)
23. Schultz, J., Milpetz, F., Bork, P., Ponting, C.P.: SMART, a simple modular architecture research tool: identification of signaling domains. Proc. Natl. Acad. Sci. USA 95, 5857–5864 (1998)
24. MOTIF Search, http://motif.genome.jp/
25. Hughes, J.D., Estep, P.W., Tavazoie, S., Church, G.M.: Computational identification of cis-regulatory elements associated with groups of functionally related genes in Saccharomyces cerevisiae. J. Mol. Biol. 296, 1205–1214 (2000)
26. Neduva, V., Linding, R., Su-Angrand, I., Stark, A., de Massi, F., Gibson, T.J., Lewis, J., Serrano, L., Russell, R.B.: Systematic discovery of new recognition peptides mediating protein interaction networks. PLoS Biol. 3, e405 (2005)
27. Favorov, A.V., Gelfand, M.S., Gerasimova, A.V., Ravcheev, D.A., Mironov, A.A., Makeev, V.J.: A Gibbs sampler for identification of symmetrically structured, spaced DNA motifs with improved estimation of the signal length. Bioinformatics 21, 2240–2245 (2005)
28. Banerjee, N., Chidambarathanu, N., Michael, D., Balakrishnan, N., Sekar, K.: An Algorithm to Find All Identical Internal Sequence Repeats. Curr. Sci. 95, 188–195 (2008)
29. Sorek, R., Kunin, V., Hugenholtz, P.: CRISPR - a widespread system that provides acquired resistance against phages in bacteria and archaea. Nat. Rev. Microbiol., 181–186 (2008)
30. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. J. Mol. Biol. 215, 403–410 (1990)
31. Thompson, J.D., Higgins, D.G., Gibson, T.J.: CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. Nucl. Acids Res. 22, 4673–4680 (1994)

Discovery of Non-induced Patterns from Sequences

Andrew K.C. Wong, Dennis Zhuang, Gary C.L. Li, and En-Shiun Annie Lee

Department of System Design University of Waterloo,
200 University Avenue West, Waterloo, Ontario, Canada
akcwong@pami.uwaterloo.ca, dennizeh@gmail.com,
gclli@pami.uwaterloo.ca, ealee@engmail.uwaterloo.ca

Abstract. Discovering patterns from sequence data has significant impact in genomics, proteomics and business. A problem commonly encountered is that the patterns discovered often contain many redundancies resulted from fake significant patterns induced by their strong statistically significant subpatterns. The concept of statistically induced patterns is proposed to capture these redundancies. An algorithm is then developed to efficiently discover non-induced significant patterns from a large sequence dataset. For performance evaluation, two experiments were conducted to demonstrate a) the seriousness of the problem using synthetic data and b) top non-induced significant patterns discovered from Saccharomyces cerevisiae (Yeast) do correspond to the transcription factor binding sites found by the biologists. The experiments confirm the effectiveness of our method in generating a relatively small set of patterns revealing interesting, unknown information inherent in the sequences.

Keywords: Sequence Pattern Discovery, Statistically Induced Patterns, Suffix Tree.

1 Introduction

Sequence data is a very significant type of data in many forms: biological sequence, web click stream, custom purchase history, event sequence, etc. A vast amount of such data from the genomic, proteomic and business arenas has been collected. The discovery of new interesting knowledge from these data has important applications and great value.

Many approaches have been developed to discover patterns from sequences. One common problem encountered is that the quality of the output patterns is overlooked resulting in an overwhelming number of output patterns [1]. To reduce the output size, some methods [1] [2] identify the redundancy among output patterns and discover those irredundant patterns. Others [3] [4] [5] [6] use statistical hypothesis test to extract and rank statistically significant patterns based on how much the frequency of a pattern deviates from the expected one by assuming a background random model. It is hoped that patterns occurring with significantly higher frequency will correspond to the functional units inherent in the sequences. However, some of them are fake or statistically redundant patterns which are considered as significant merely because they contain very strong subpatterns [7]. This problem is exaggerated in dense datasets containing many strong patterns.

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 149–160, 2010.

In this paper, we present the concept of statistically induced patterns to caputure these fake patterns and an efficient algorithm based on generalized suffix tree to discover statistically non-induced patterns. By removing induced patterns, the quality of output patterns can be further improved and the ranking of important functional units can be elevated. Our method is scalable to handle very large sequence data rendering a more compact ouput. Though our method provides a general data mining framework, here we focus specifically on biological data (transcription binding site data).

2 Related Work

Pattern discovery techniques or motif-finding algorithms have evolved in a fast pace in bioinformatics. This is driven by the rapid growth of available DNA and protein sequence database as well as the strong desire to find functional units such as regulation signals embedded in biological sequences. Pattern discovery techniques are developed to reveal such conserved patterns across sequences. In motif finding, two main perspectives are adopted: the probabilistic and the combinatorial. The former uses the profile-based position weight matrix (PWM) to find the location of the motifs in the sequences [8] [9]. Thus, the best motif is the most probable PWM. In the latter, a motif is defined as a consensus that occurs repeatedly in sequences [3] [4] [5] [6] [10]. The problem of producing overwhelming number of patterns is often encountered in the latter approaches.

Extracting statistically significant patterns is one way of shrinking output size. A linear time algorithm is presented in [4] to detect statistically significant patterns (overrepresented δ-significant patterns) which are represented by the internal nodes of the suffix tree. Statistics such as mean and variance are efficiently annotated to each node. Observing that not all nodes in the suffix tree correspond to overrepresented δ-significant pattern, we develop a method to identify those that are not.

Although extracting only statistically significant patterns greatly reduces the output size, often these patterns form a large set of patterns with real motifs mixed with many of their random variations [7]. A background model of order 3 Markov chain and a greedy algorithm have been proposed to separate the artifacts from the real motifs. Our definition of statistically induced patterns is a variation of Blanchette and Sinha's [7] artifact motifs. However, ours uses the Bernoulli scheme instead of markov chain and hence does not require training a complex Markov model. Furthermore, our method in discovering non-induced patterns is more efficent.

3 Methodology

3.1 Preliminary Definitions

Let Σ be a set of distinct elements $\{e_1, e_2, \ldots, e_{|\Sigma|}\}$, called the alphabet, and $|\Sigma|$ be its size. A sequence S over Σ is an ordered list of elements $s_1 s_2 \ldots s_n$. A pattern P is a short sequence $p_1 p_2 \ldots p_m$ over Σ and $|P|$ is its order. We call P a consecutive pattern with no gaps. In general, the input data might come as multiple sequences $S_1, S_2, \ldots, S_N$ with lengths $l_1, l_2, \ldots, l_N$ respectively. Let L be their overall length.

The *number of occurrences* of P in multiple sequences is denoted by k_P. The list of occurrence positions is $L_P = \{\ldots, (i,j), \ldots\}$ where the ordered pair (i,j) is a position denoting that P occurs at position j in sequence i. The *support* of P denoted by q_P in the multiple sequences is the number of sequences in which P occurs at least once. A pattern is said to be *frequent* if its number of occurrences is not less than a specified minimum requirement min_{occ}. Mathematically, that is $k_p \geq min_{occ}$.

3.2 Statistical Model for Input Sequences

A background random model for determining the expected frequency of P is needed to define statistically significant patterns. Without being given specific domain knowledge for the background model, we adopt a simplest model: the Bernoulli scheme. With this scheme, the probability of a pattern P occurring in a position of a random sequence is $pr(P) = \prod_{i=1}^{m} pr(p_i)$, where $p_i \in \Sigma$. Let X_i be a Bernoulli variable that indicates whether P occurs in position i of a random sequence. The total number of possible positions is $T_P = \sum_{i=1}^{n}(l_i - m + 1)$, so the number of occurrences of P is a random variable $X_P = \sum_i X_i$ which follows a binominal distribution. Its expected number of occurrences is $E(X_P) = pr(P) \cdot T_P$.

Definition 1. Statistically significant pattern
To measure how k_P of P deviates from its expected frequency if the given sequences are generated from the random model, we use the *standard residual* [11]

$$z_P = \frac{k_P - E(X_P)}{\sqrt{E(X_P)}}$$

A pattern is *statistically significant or overrespresented* [12] if $z_P \geq t$ where t is the predefined minimum threshold.

Definition 2. Significant representative pattern
As observed in the paper [4] [16], patterns can be clustered into equivalence groups $C_1, C_2, ..., C_K$ such that patterns in the same group C_i have the same list of occurrence positions L_P.

Representative pattern is the pattern in the group C_i that has the highest statistical significance z_P or equivalently has the highest order. *Significant representative pattern* is both statistically significant and representative.

Definition 3. Statistically induced pattern
Let P' be a subpattern of P. The *conditional statistical significance* of P given P' is defined as

$$z_{P|P'} = \frac{k_P - E(X_P|P')}{\sqrt{E(X_P|P')}}$$

where $E(X_P|P') = pr(P|P') \cdot k_{P'} = \frac{pr(P)}{pr(P')} \cdot k_{P'}$.

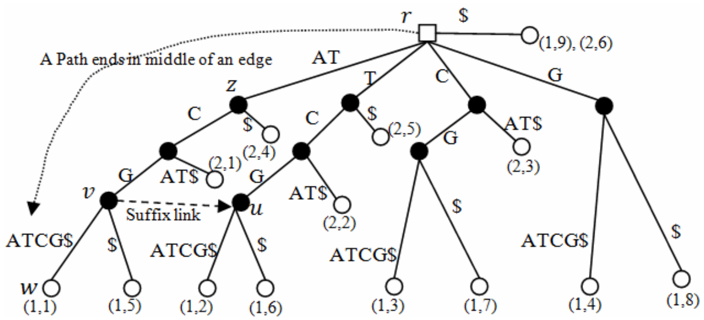


Fig. 1. Generalized suffix tree T for multiple strings $S_1 = \mathrm{ATCGATCG\$}$ and $S_2 = \mathrm{ATCAT\$}$. The square node is the root, the solid circles are the internal nodes and the hollow circles denote the leaf nodes. r, u, v, w and z are the nodes in the suffix tree. Edges are labelled with substrings. The dotted arrow shows the end point "T" for a path ending inside an edge. $(1, 1)$ is a position.

Given a set of significant representative patterns, a pattern P in it is said to be *statistically induced* if there exists a *proper* subpattern P' of P such that $z_{P|P'} < t$. A proper subpattern P' is not statistically induced.

This conditional statistical significance is used to evaluate how strongly the statistical significance of a pattern is attributed by the occurrences of one of its proper subpatterns. Those induced patterns whose significances are due to their proper subpatterns by mere chance are fake patterns. Hence removing them would render a more succinct set of patterns.

3.3 Characterizing Significant Representative Patterns in Generalized Suffix Tree

First, we introduce the generalized suffix tree as the data structure for representing strings. It can be constructed in $O(L)$ time and space. The details of it and its linear time and space construction algorithms can be found in [13]. Here we establish the connection between consecutive patterns and path labels in the suffix tree. Finally, we link significant representative patterns with nodes in the suffix tee.

Generalized Suffix Tree

Given a collection of strings $S_1, S_2, \ldots, S_N$ over Σ, the generalized suffix tree T for these multiple strings is a rooted directed tree with the following properties:

(1) Each leaf node is labelled by a set of positions $\{\ldots, (i, j), \ldots\}$ where (i, j) indicates a suffix of string S_i starting at the position j.

(2) Each internal node has at least two outgoing edges each of which is labelled with a non-empty substring of one of the input string. No two edges out of a node can have the edge-label starting with the same character.

Most often, a termination character $\$ \notin \Sigma$ is appended to each string to ensure that T exists for this set of multiple strings. Fig. 1 gives an example for two input strings.

Consecutive Pattern in Suffix Tree

The label of the path from root ending at node v is the string resulted from concatenation of the substrings that label the edges along that path. The label of a path from root ending inside an edge (v, w), is the label of the path from the root ending at node v, concatenating with the remaining characters of the label of the edge (v, w) down to the end of the path. For convenience, the label of a path ending at node v is represented by $pl(v)$, the path label of v. In Fig. 1, the path label of node u is the substring TCG of S_1. The label of a path ending in the middle of the edge (v, w) with end point indicated as "T" is the substring ATCGAT of S_1. Accordingly, a consecutive pattern that occurs at least once in the input strings has its unique path in suffix tree and is represented by the path label.

Frequent Pattern in Suffix Tree

The number of occurrences of a consecutive pattern is the number of positions found under its path in the suffix tree. For example, positions $\{(1,1),(1,5),(2,1)\}$ are found under the path of pattern ATC. By storing into each node x the number of positions $k(x)$ in the subtree rooted by it, the number of occurrences of a consecutive pattern can be easily obtained by finding the node at or above which its path ends. For example, the number of occurrences of pattern TCG whose path ends at u is given by $k(u) = 2$. Hence frequent patterns are represented as labels of paths that end at or above a node x where $k(x) >= min_{occ}$.

Representative Pattern in Suffix Tree

Note that the paths of representative patterns end at nodes instead of within edges. A pattern with path ending within an edge can be further extended by at least one character to the right without decreasing the number of occurrences and thus by definition cannot be a representative pattern. For example, the pattern A, which ends inside the edge (r, z), has a superpattern AT ending at node z with the same number of occurrences indicated by $k(z) = 4$.

However, it is not a one-to-one mapping; not all nodes correspond to representative patterns. As we might notice that the pattern associated with the path ending at the node u in Fig. 1, $pl(u) = \mathrm{TCG}$, is not a representative pattern because it has a superpattern $pl(v) = \mathrm{ATCG}$ with the same number of occurrences as indicated by $k(u) = k(v) = 2$ which has higher statistical significance. Hence, we need to identify nodes corresponding to representative patterns in the tree. This can be efficiently achieved by utilizing the suffix links. A suffix link of v points to u if $pl(v)$ is an one character left extension of $pl(u)$. The node u is called the suffix node of v because $pl(u)$ is the suffix of $pl(v)$. For example, $pl(v) = \mathrm{ATCG}$ is a string by appending A to the left of $pl(u) = \mathrm{TCG}$. Only one suffix link is shown in Fig. 1.

In summary, a representative pattern corresponds to a node x in a suffix tree that is not a suffix node of the other node. A representative pattern $pl(x)$ is statistically significant if $z_{pl(x)} \geq t$. For example, $\{\mathrm{ATCG}, \mathrm{TCG}, \mathrm{CG}\}$ forms an equivalence set and the representative pattern is ATCG and hence corresponds to the node v that does not have suffix link pointing to it.

3.4 Discovering Non-induced Patterns

Although significant representative patterns contain fewer redundancies, they could still be too many for human experts to interpret. As noted in **Definition 3**, some significant representative patterns can be statistically induced by others and should be removed. Here we describe an efficent algorithm to find non-induced patterns.

If pattern P is not induced by the proper subpattern P' that has the smallest conditional statistical significance $z_{P|P'}$ among all proper subpatterns of P, where we denotes P' as the valid pattern for P, then P is not statistically induced. In other word, if P is non-induced, then $z_{P|P'} >= t$ for any proper subpattern P' of it, including the one with smallest $z_{P|P'}$. Thus, we develop **Algorithm 1** to efficiently discover non-induced patterns by identifying the valid pattern for each significant representative pattern from the lowest to highest order. Note that each representative pattern corresponds to a node in the suffix tree.

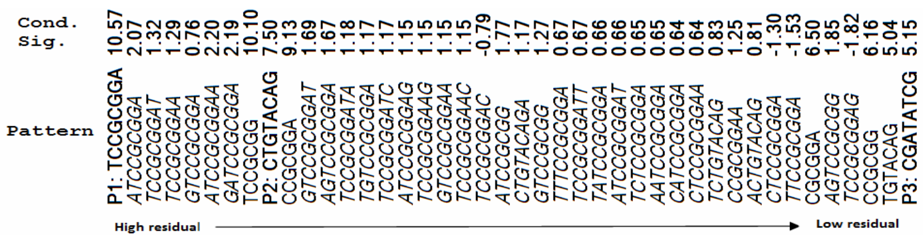
Algorithm 1. Discovery of non-induced patterns

```
1.  Construct a generalized suffix tree $T$ for the input sequences
2.  Annotate $k(v)$ the number of positions under each node $v$ of $T$
3.  Extract a set of nodes whose $k(v) \geq min_{occ}$
4.  Sort the above nodes in ascending order according to order of $pl(v)$ using counting sort.
5.  For each node $v$
      a)Find the valid node $w$ for $v$ using **Procedure 1**
      b)If $v$ is not a suffix node and $z_{pl(v)} \geq t$ and $pl(v)$ is not induced by $pl(w)$
          Output $pl(v)$
        End if
    End for
```

Procedure 1. Find valid node for v

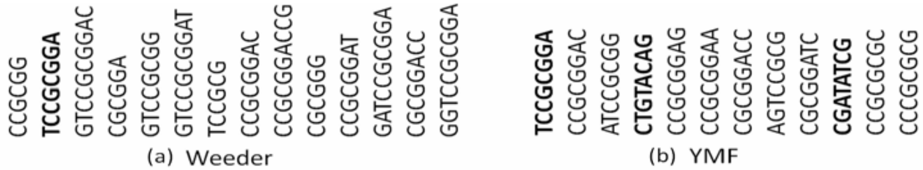
```
1.  Let $v_S$ and $v_P$ be the suffix node and parent node of $v$ respectively
2.  If $pl(v_S)$ is non-induced
        Let $w_1$ be $v_S$
3.  Else
        Let $w_1$ be the valid node of $v_S$
    End if
4.  If $pl(v_P)$ is non-induced
        Let $w_2$ be $v_P$
5.  Else
        Let $w_2$ be the valid node of $v_P$
    End if
6.  Pick one node with the smallest conditional statistical significance out of $w_1$ and $w_2$ to be
    the proper node of $v$
```

Running time analysis for Algorithm 1. Step 1-3 can be achieved in linear time. Step 4 uses counting sort to sort the nodes according to the path length and can thus be done also in linear time. Steps 5a and 5b take constant time. Step 5 can be done in $O(L)$ time. Therefore, non-induced patterns can be found in linear time.



Pattern	Cond. Sig.
P1: TCCGCGGA	10.57
ATCCGCGGA	2.07
TCCGCGGAT	1.32
TCCGCGGAA	1.29
GTCCGCGGA	0.76
ATCCGCGGAA	2.20
GATCCGCGGA	2.19
TCCGCGG	10.10
P2: CTGTACAG	7.50
CCGCGGA	9.13
GTCCGCGGAT	1.69
AGTCCGCGGA	1.67
TCCGCGGATA	1.18
TGTCCGCGGA	1.17
TCCGCGGATC	1.17
ATCCGCGGAG	1.15
TCCGCGGAAG	1.15
GTCCGCGGAA	1.15
TCCGCGGAAC	1.15
TCCGCGGAC	-0.79
ATCCGCGG	1.77
CTGTACAGA	1.17
GTCCGCGG	1.27
TTTCCGCGGA	0.67
TCCGCGGATT	0.67
TATCCGCGGA	0.66
ATCCGCGGAT	0.66
TCTCCGCGGA	0.65
AATCCGCGGA	0.65
CATCCGCGGA	0.64
CTCCGCGGAA	0.64
TCTGTACAG	0.83
CCGCGGAA	1.25
ACTGTACAG	0.81
CTCCGCGGA	-1.30
TTCCGCGGA	-1.53
CGCGGA	6.50
AGTCCGCGG	1.85
TCCGCGGAG	-1.82
CCGCGG	6.16
TGTACAG	5.04
P3: CGATATCG	5.15

Fig. 2. Patterns reported by Method 1. Patterns ranked higher than P_3 by standard residual are shown. Cond. Sig. is the short form of conditional statistical significance.



CCGCGG
TCCGCGGA
GTCCGCGGAC
CGCGGA
GTCCGCGG
GTCCGCGGAT
TCCGCG
CCGCGGAC
CCGCGGACCG
CGCGGG
CCGCGGAT
GATCCGCGGA
CGCGGACC
GGTCCGCGGA

(a) Weeder

TCCGCGGA
CCGCGGAC
ATCCGCGG
CTGTACAG
CCGCGGAG
CCGCGGAA
CGCGGACC
AGTCCGCG
CGCGGATC
CGATATCG
CCGCGCGC
CCCGCGCG

(b) YMF

Fig. 3. Highest ranking patterns discovered by Weeder and YMF. The implanted patterns are bolded.

4 Experimental Results

Two sets of experiments were conducted for performance evaluation. In both experiments, min_{occ} is set to 5 and the statistical threshold t is set to 3.

4.1 Experiment on Synthetic Data

To show that our method could remove statistically induced patterns and raise the ranking of the truly significant patterns, 100 random sequences of 1000 bases over DNA alphabet are created. Three strong patterns $P_1 = \mathrm{TCCGCGGA}$, $P_2 = \mathrm{CTGTACAG}$ and $P_3 = \mathrm{CGATATCG}$ are implanted into these sequences such that their standard residuals z_1, z_2 and z_3 are 48, 24, and 12 respectively. We apply Method 1 to discover significant representative patterns and Method 2 to obtain non-induced patterns.

Fig. 2 shows the patterns of P_1, P_2 and P_3 and their superpatterns and subpatterns which are ranked higher than P_3 according to their standard residual. Those patterns in italic font are superpatterns induced by P_1, P_2 and P_3 (indicated by their conditional statistical significance, i.e. less than the prescribed threshold of 3), hence they are removed by method 2. After these induced patterns are removed, the ranking of P_3 according to the standard residual is raised from the 42th to the 7th. The number of patterns reported by method 1 is 527 while the number of patterns reported by method 2 is reduced to 315 with a 40.2% reduction rate. Hence, as anticipated, a more compact set of patterns is obtained by ensuring that patterns are non-induced and the rankings of the real significant patterns are elevated.

Two well known motif finding tools YMF [3] and Weeder [10] are applied to the synthetic data respectively for comparison. For Weeder, medium mode is selected. For YMF, no spacers and degenerate symbols are allowed and the motif length is fixed to 8. Both methods require specifying the organisim from which the input sequences are obtained. Since the synthetic data does not come from any real organisim, we choose an arbitary organism Saccharomyces cerevisiae (SC) for both methods. The highest ranking patterns discovered by Weeder and YMF are shown in Fig. 3. Most discovered patterns are related to the strong implanted pattern P_1 and can be considered as induced patterns. P_1 is among the top 15 patterns from Weeder while YMF discovered all implanted patterns P_1, P_2 and P_3 within top 10 patterns. Note that YMF has the advantage position by knowing the pattern length in advance. Compared to Weeder and YMF, our method is more general: (1) it searches for patterns with arbitrary length while pattern length is restricted from 6 to 10 for the other two; (2) it does not require background information while the others require specifiying the input organisim. In other words, YMF and Weeder are designed more specifically for TF binding site discovery.

4.2 Experiment on Transcription Factor Binding Sites

We next examine the capability of our method in discovering biological functional units, which is the foremost fundamental step towards understanding the complex mechanism of the gene expression regulations. We apply our method to identify transcription factor (TF) binding sites on Yeast using the widely studied SCPD database [14] with many of its TFs known along with their regulated genes. They are from the upstream (promoter) regions of genes regulated by one or more TFs. Each set of genes is called regulon and is associated with one or more TFs. The genes are believed to be co-regulated by specific TFs and the binding sites for them are experimentally determined in the database. Three conditions are imposed when choosing the regulon: (1) the number of genes in it should be at least three, (2) the consensus binding sites are available, and (3) the number of gaps or “don’t care” characters in the consensus should be at most two. The condition (3) is imposed since we discover only consecutive patterns in the current stage. There are totally 18 such regulons. For each regulon of the TF(s), the upstream sequences of genes are extracted from position +1 to -800 relative to the ORF (translation start site).

We design a score combining the statistical significance and support to rank the discovered patterns since the former is based only on its number of occurrences and no information of its support is used. However, to find transcription binding sites amongst multiple sequences, the number of supports is important. These genes are regulated by one or more TFs, and thus we expect that each upstream region of the input gene sequence contains one or more binding sites. Hence, patterns with higher support should be considered more important than those with less support. For example, if we have discovered two patterns TTTAAA and CTTCCT with close statistical residual but different support, say 2 and 7, then the latter will be more important and more likely to correspond to binding sites. Hence, a combined score is defined as $score = \frac{\text{Support}}{\text{No. of genes}} \cdot \text{Standard residual}$.

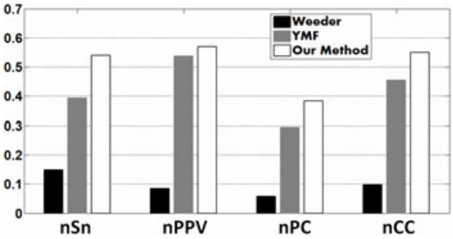


Fig. 4. Combined measures over 18 datasets

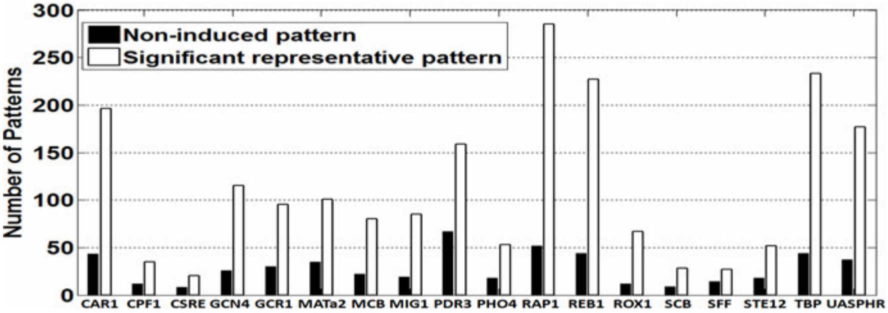


Fig. 5. Number of reported patterns after removing induced patterns among significant representative patterns

In DNA sequences tandem, repeats are common. For example, in a sequence like AAAAAATTTTTT, the pattern AAAA occurs at positions 1, 2 and 3 which overlap multiple times. Hence, a post-processing step is applied to further remove patterns whose occurrences overlap in the original sequences.

We discovered the non-induced patterns for each dataset and compared the result with YMF and Weeder respectively (Table 1). We ranked the discovered patterns according to the combined score and chose the top 15 ones for comparision. For YMF, we used its webserver and obtained 5 best motifs through FindExplanators [7] for motif length from 6 to 8 (all available parameters), resulting a total number of 15 patterns. 0 spacers and 2 degnerate symbols are allowed in the motif definition. For Weeder, we downloaded the standalone platform and used the medium mode. All motifs recommend in the final output are used for comparison. For each motif reported by Weeder, we use only the best occurrences with the percentage threshold greater than 90 as binding site predictions. We use the measures nSn (sensitivity), nPPV (positive preditive value), nPC (performance coefficient) and nCC (correlation coefficient) defined in [15] in comparison.

Among the 18 datasets, our discovered patterns within rank 13 match the consensus binding sites in 14 datasets and 4 of them are ranked top. The patterns in bold do not match the binding sites in the remaining 4 datasets CPF1, CSRE, MATalpha2 and SFF. The reason why our discovered patterns have no match in CPF1, CSRE and SFF is that their consensus binding sites have fewer than 2 occurrences.

As for MATalpha2, the consensus has 6 occurrences, but it has many substitutions. Because our program runs with $min_{occ} = 5$ and is confined to discovering consecutive patterns, these consensuses are not discovered.

The overall performances of Weeder, YMF and our method across 18 datasets are evaluated by the combined measures in [15] and shown in Fig. 4. It indicates that the overall perfromance of our method is better than YMF. Weeder does not perform well comparatively and the reason might be that the percentage threshold 90 is too strict. However, Weeder does not provide a good strategy in choosing this parameter.

Fig. 5 shows the number of reported patterns in terms of non-induced pattern and significant repressentative patterns. After removing induced patterns among significant repressentative patterns, our method produces a relative small set of patterns of which the number of reported patterns ranges from 8 to 67. The result shows that our method is able to retain those patterns associated with conserved functional units in the promoter regions while reducing the number of patterns.

Table 1. Comparison of our method, YMF and Weeder on SPCD datasets (the pattern among the top 15 that achieves the best nSn is used for comparison). IUPAC Nucleotide Code is used.

TF		Motif/Pattern	nSn	nPPV	nPC	nCC	Rank
CAR1	Consensus	AGCCGCCR					
	Weeder	CCTAGCCG	0.23	0.09	0.07	0.14	
	YMF	GCCGCCG	0.7	1	0.7	0.84	
	Our Method	AGCCGCC	0.88	1	0.88	0.94	6
CPF1	Consensus	TCACGTG					
	Weeder	CACGTGGC	0	0	0	-0.01	
	YMF	YCACGWG	1	0.5	0.5	0.71	
	Our Method	**TTC**	**0.29**	**0.01**	**0.01**	**0.04**	**10**
CSRE	Consensus	YCGGAYRRAWGG					
	Weeder	GCGGTCGG	0	0	0	-0.01	
	YMF	CGGATRRA	0.58	0.22	0.19	0.35	
	Our Method	**CCGG**	**0.33**	**0.08**	**0.07**	**0.15**	**1**
GCN4	Consensus	TGANT					
	Weeder	TGACTC	0.07	0.13	0.05	0.08	
	YMF	TGWCTR	0.18	0.51	0.15	0.29	
	Our Method	TGACT	0.34	1	0.34	0.57	13
GCR1	Consensus	CWTCC					
	Weeder	TCTGGCATCC	0.1	0.2	0.07	0.13	
	YMF	TCTYCCY	0.3	0.48	0.23	0.37	
	Our Method	TTCC	0.68	0.39	0.33	0.5	9
MATalpha2	Consensus	CRTGTWWWW					
	Weeder	GGAAATTTAC	0.13	0.14	0.07	0.13	
	YMF	ACGCGT	0	0	0	0	
	Our Method	**GAAAAAAG**	**0**	**0**	**0**	**-0.01**	**1**
MCB	Consensus	WCGCGW					
	Weeder	AGACGCGT	0.19	0.08	0.06	0.1	
	YMF	ACGCGT	0.68	1	0.68	0.82	
	Our Method	ACGCGT	0.68	1	0.68	0.82	1
MIG1	Consensus	CCCCRNNWWWWW					
	Weeder	CCCCAG	0.39	0.1	0.09	0.19	
	YMF	CCCCRS	0.5	0.21	0.18	0.32	
	Our Method	CCCCAG	0.33	0.29	0.18	0.3	2
PDR3	Consensus	TCCGYGGA					
	Weeder	GTCTCCGCGG	0.32	0.14	0.11	0.19	
	YMF	TCCGYGGA	1	1	1	1	

Table 1. (*continued*)

	Our Method	TCCGCGGA	0.64	1	0.64	0.8	1
PHO4	Consensus	CACGTK					
	Weeder	GAAACGTG	0.07	0.02	0.02	0.02	
	YMF	CACGTGSR	0.71	0.75	0.58	0.73	
	Our Method	CACGTG	0.71	1	0.71	0.84	1
RAP1	Consensus	RMACCCA					
	Weeder	AGCACCCA	0.13	0.23	0.09	0.17	
	YMF	CACCCA	0.64	0.86	0.58	0.74	
	Our Method	CACCCA	0.64	0.86	0.58	0.74	8
REB1	Consensus	YYACCCG					
	Weeder	ACCCGC	0.14	0.05	0.04	0.08	
	YMF	TTACCCG	0.7	1	0.7	0.84	
	Our Method	TTACCCG	0.7	1	0.7	0.84	7
ROX1	Consensus	YYNATTGTTY					
	Weeder	CCTATTGT	0.28	0.05	0.04	0.07	
	YMF	TTGTTS	0.48	0.29	0.22	0.35	
	Our Method	ATTGTT	0.6	0.63	0.44	0.6	6
SCB	Consensus	CNCGAAA					
	Weeder	AGTCACGAAA	0.47	0.26	0.2	0.31	
	YMF	CACGAA	0.61	1	0.61	0.78	
	Our Method	CACGAAA	0.71	1	0.71	0.84	1
SFF	Consensus	GTMAACAA					
	Weeder	CTGTTTAG	0.13	0.02	0.02	0.04	
	YMF	TAAWYA	0.38	0.08	0.07	0.17	
	Our Method	**AAAGG**	**0.13**	**0.04**	**0.03**	**0.06**	**2**
STE12	Consensus	TGAAACA					
	Weeder	ATGAAACA	0.2	0.05	0.04	0.07	
	YMF	ACATGS	0.06	0.1	0.04	0.07	
	Our Method	TGAAAC	0.86	0.7	0.63	0.77	3
TBP	Consensus	TATAWAW					
	Weeder	CCGCTG	0	0	0	-0.02	
	YMF	CRCATR	0.01	0.02	0.01	0	
	Our Method	ATATAAA	0.43	0.89	0.41	0.62	13
UASPHR	Consensus	CTTCCT					
	Weeder	TGTCAGCG	0	0	0	-0.01	
	YMF	CCTCGTT	0.14	0.21	0.09	0.17	
	Our Method	CTTCCTC	0.71	0.86	0.64	0.78	9
Average	Weeder		0.16	0.09	0.05	0.09	
	YMF		0.48	0.51	0.37	0.47	
	Our Method		0.54	0.65	0.44	0.56	

5 Conclusion and Future Work

This paper presents an efficient algorithm to discover non-induced patterns from a large sequence data. It uses a generalized suffix tree to assist the identification of significant representative patterns and the removal of the induced patterns whose statistical significance is due to their strong subpatterns. By ensuring that each pattern discovered is non-induced, our method produces a more compact pattern set.

The results from TF binding sites experiment confirm the algorithm's ability to acquire a relatively small set of patterns that reveal interesting, unknown information inherent in the sequences. While the algorithm drastically reduces the number of patterns, it is still able to retain patterns associated with conserved functional units in the promoter regions without relying on prior knowledge.

Our future work will advance in the following directions: (1) Extending our method to discover patterns with gaps; (2) Discovering distance patterns in protein sequences and relating the discovered patterns to three-dimensional conformation and low sequence similarity.

Acknowledgments. The work was supported by grants from NSERC.

References

1. Rigoutsos, I., Floratos, A.: Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm. Bioinformatics 14(1), 55–67 (1998)
2. Parida, L., Rigoutsos, I., Floratos, A., Platt, D., Gao, Y.: Pattern Discovery on Character Sets and Real-valued Data: Linear Bound on Irredundant Motifs and an Efficient Polynomial Time Algorithm. In: Proceedings of the eleventh ACM-SIAM Symposium on Discrete Algorithms, pp. 297–308 (January 2000)
3. Sinha, S., Tompa, M.: Discovery of novel transcription factor binding sites by statistical overrepresentation. Nucleic Acids Research 30(24), 5549–5560 (2002)
4. Apostolico, A., Bock, M., Lonardi, S., Xu, X.: Efficient Detection of Unusual Words. Journal of Computational Biology 7(1/2), 71–94 (2000)
5. Eskin, E., Pevzner, P.: Finding composite regulatory patterns in DNA sequences. Bioinformatics 18(suppl. 1), S354–S363 (2002)
6. Marsan, L., Sagot, M.: Extracting structured motifs using a suffix tree - Algorithms and application to promoter consensus identification. Journal of Computational Biology 7(3/4), 345–362 (2000)
7. Blanchette, M., Sinha, S.: Separating real motifs from their artifacts. Bioinformatics 17(suppl. 1), S30–S38 (2001)
8. Sze, S., Lu, S., Chen, J.: Integrating Sample-Driven and Pattern-Driven Approaches in Motif Finding. In: Algorithms in Bioinformatics: 4th International Workshop, pp. 438–449 (2004)
9. Bailey, T.L., Elkan, C.: Unsupervised learning of multiple motifs in biopolymers using expectation maximization. Machine Learning 21, 51–80 (1995)
10. Pavesi, G., Zambelli, F., Pesole, G.: WeederH: an algorithm for finding conserved regulatory motifs and regions in homologous sequences. BMC Bioinformatics 8, 46 (2007)
11. Haberman, S.: The Analysis of Residuals in Cross-Classified Tables. Biometrics 29, 205–220 (1973)
12. Wong, A., Wang, Y.: High-Order Pattern Discovery from Discrete-Valued Data. IEEE Trans on Knowledge Systems 9(6), 877–893 (1997)
13. Gusfield, D.: Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology (1997)
14. SCPD database, `http://rulai.cshl.edu/SCPD/`
15. Tompa, M., Li, N., Bailey, T., Church, G., Moor, B., Eskin, E., Favorov, A., Frith, M., Fu, Y., Kent, W., Makeev, V., Mironov, A., Noble, W., Pavesi, G., Pe-sole, G., Regnier, M., Simonis, N., Sinha, S., Thijs, G., Helden, J., Vandenbogaert, M., Weng, Z., Workman, C., Ye, C., Zhu, Z.: Assessing Computational Tools for the Discovery of Transcription Factor Binding Sites. Nature Biotechnology 23(1), 137–144 (2005)
16. Wong, A., Li, G.: Simultaneous Pattern Clustering and Data Grouping. IEEE Trans. Knowledge and Data Engineering 20(7), 911–923 (2008)

Exploring Homology Using the Concept of Three-State Entropy Vector

Armando J. Pinho[1], Sara P. Garcia[1], Paulo J.S.G. Ferreira[1], Vera Afreixo[2],
Carlos A.C. Bastos[1], António J.R. Neves[1], and João M.O.S. Rodrigues[1]

[1] Signal Processing Lab, DETI / IEETA,
University of Aveiro, 3810–193 Aveiro, Portugal
[2] Department of Mathematics,
University of Aveiro, 3810–193 Aveiro, Portugal
{ap,spgarcia,pjf,vera,cbastos,an,jmr}@ua.pt

Abstract. The three-base periodicity usually found in exons has been used for several purposes, as for example the prediction of potential genes. In this paper, we use a data model, previously proposed for encoding protein-coding regions of DNA sequences, to build signatures capable of supporting the construction of meaningful dendograms. The model relies on the three-base periodicity and provides an estimate of the entropy associated with each of the three bases of the codons. We observe that the three entropy values vary among themselves and also from species to species. Moreover, we provide evidence that this makes it possible to associate a three-state entropy vector with each species, and we show that similar species are characterized by similar three-state entropy vectors.

Keywords: DNA signature, DNA coding regions, DNA entropy, Markov models.

1 Introduction

It is well-known that there are periodicities in DNA sequences, the strongest of which is generally associated with the period three that can be found in the exons of prokaryotes and eukaryotes [14,2]. This three-base periodicity has been used, for example, for predicting potential protein-coding regions [4,13,6,7,15] and for finding potential reading frame shifts in genes [5].

In a previous work [3,9], we have used this property for exploring the possibility of using a three-state finite-context model with the aim of improving the compression of the protein-coding regions of the DNA sequences. That study led us to the conclusion that, for those protein-coding DNA regions, a model that switches sequentially between three states provides better compression than a model based on a single state. Moreover, the three-state model looses its efficacy when applied to unrestricted DNA sequences, which provides additional evidence towards the distinctive three-base periodicity of the protein-coding regions.

Besides the observation that a three-state finite-context model works better than a single-state model in protein-coding regions, we also observed a phenomenon that caught our attention. Each of the three states of the finite-context

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 161–170, 2010.

model can be viewed as a model of the information source associated to each of the three nucleotides that form a codon. Since we are able to estimate the entropy of each of the three states of our model, we are also able to estimate the average information carried out by each of the three nucleotides. The interesting finding that we have made was that both the absolute and the relative values of these entropies vary among the species [3,9]. In other words, the average information conveyed when the first, second or third bases of a codon are specified is not the same, and the differences depend on the species.

In this paper, we further investigate this phenomenon and, particularly, we try to find out if the differences among the values of the entropies of the three base positions of the codon could be used as a species signature. Although still preliminary, the results obtained suggest that this is in fact true, i.e., that we are able to construct a low-dimensional entropy vector capable of correctly clustering similar species. Therefore, these findings may contribute to the development of new methods for alignment-free sequence comparison.

2 Materials and Methods

2.1 DNA Sequences

In this preliminary study, we used thirteen species, nine eukaryotes (five animals and four plants) and four prokaryotes (bacteria), listed in Table 1. When available, we used the RNA data provided in a single file. In the other cases, we used the data of the ".ffn" files. In the case of the *Ricinus communis* we used the ".cds" data. Because the performance of the three-state model is affected by losses of synchronization in the reading frame, i.e., it assumes that, for example, the first base of the codon is always handled by state one of the model, we only considered sequences whose length is a multiple of three and that do not contain undefined symbols. Moreover, for these experiments, and also with the aim of avoiding inconsistencies in the expected codon structure, we did not consider those that do not start with ATG.

2.2 Finite-Context Models

Consider an information source that generates symbols, s, from the alphabet $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$. Consider that the information source has already generated the sequence of n symbols $x^n = x_1 x_2 \ldots x_n$, $x_i \in \mathcal{A}$. A finite-context model (see Fig. 1) assigns probability estimates to the symbols of the alphabet, regarding the next outcome of the information source, according to a conditioning context computed over a finite and fixed number, $k > 0$, of the most recent past outcomes $c = x_{n-k+1} \ldots x_{n-1} x_n$ (order-k finite-context model) [1,10,11]. Therefore, the number of conditioning states of the model is 4^k.

The probability estimates, $P(X_{n+1} = s|c), \forall_{s \in \mathcal{A}}$, are usually calculated using symbol counts that are accumulated while the sequence is processed, which makes them dependent not only of the past k symbols, but also of n. In other words, these probability estimates will in general vary as a function of the position along the sequence.

Table 1. Organisms used in this study

Organism	Reference
Homo sapiens (human)	Build 37.1
Pan troglodytes (chimpanzee)	Build 2.1
Macaca mulatta (rhesus macaque)	Build 1.1
Mus musculus (mouse)	Build 37.1
Rattus norvegicus (brown rat)	Build 4.1
Arabidopsis thaliana (thale cress)	NC003070/1/4/5/6
Populus trichocarpa (black cottonwood)	Version 2.0
Vitis vinifera (grape vine)	Build 1.1
Ricinus communis (castor oil plant)	Release 0.1
Streptococcus pneumoniae strain ATCC 700669	NC011900
Chlamydia trachomatis strain D/UW-3/CX	NC000117
Mycoplasma genitalium strain G-37	NC000908
Streptococcus mutans strain UA159	NC004350

Typically, the probability estimates produced by the finite-context model are used to drive an arithmetic encoder, which is able to generate output bit-streams with average bitrates almost identical to the entropy of the model [1,10,11]. The theoretical bitrate average of the finite-context model after encoding n symbols is given by

$$H_n = -\frac{1}{n}\sum_{i=1}^{n}\log_2 P(X_i = x_i|c) \quad \text{bpb}, \tag{1}$$

where $c = x_{i-k}\dots x_{i-2}x_{i-1}$ and "bpb" stands for "bits per base". Recall that the entropy of any sequence of four symbols is limited to two bits per symbol, a value that is obtained when the symbols are independent and equally likely.

The probability that the next outcome, X_{n+1}, is s, where $s \in \mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$, is obtained using the estimator

$$P(X_{n+1} = s|c) = \frac{n_s^c + \alpha}{n^c + \alpha|\mathcal{A}|}, \tag{2}$$

where n_s^c represents the number of times that, in the past, the information source generated symbol s having as conditioning context $c = x_{n-k+1}\dots x_{n-1}x_n$ and where

$$n^c = \sum_{s\in\mathcal{A}} n_s^c \tag{3}$$

is the total number of events that has occurred so far in association with context c. The parameter α controls how much probability is assigned to possible but yet unseen events. In this work, we used $\alpha = 1$, which transforms the estimator into the multinomial extension of Laplace's rule of succession [8].

Note that, initially, when all counters are zero, the symbols have probability 1/4, i.e., they are assumed equally probable. The counters are updated each time a symbol is encoded. Since the context template is causal, the decoder is able to

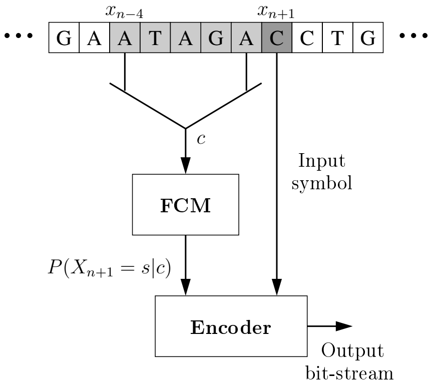


Fig. 1. Example of a finite-context model: the probability of the next outcome, X_{n+1}, is conditioned by the k last outcomes. In this example, $\mathcal{A} = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ and $k = 5$. The "Encoder" block is usually an arithmetic encoder.

reproduce the same probability estimates without needing additional information. In other words, this model is self-contained, in the sense that it is capable of recovering the original sequence based only on the bit-stream produced by the encoder.

2.3 The Three-State Model

Figure 2 shows the model addressed in this paper. It differs from the finite-context model displayed in Fig. 1 by the inclusion of three internal states. Each state is selected periodically, according to a three-base period, and comprises a finite-context model, similar to the one presented in Fig. 1.

The three-state model, originally introduced in [3,9] with the purpose of compressing protein-coding regions of DNA, is revisited in this paper with the aim of exploring homology using the idea of a three-state entropy vector.

With this model, probabilities depend not only on the k last outcomes, but also on the value of (n mod 3), which is used for state selectivity. In this case, the probability estimator is given by

$$P(X_{n+1} = s|c) = \frac{n_s^{c,\phi} + \alpha}{n^{c,\phi} + \alpha|\mathcal{A}|}, \tag{4}$$

where

$$\phi = n \bmod 3 + 1 \quad \text{and} \quad n^{c,\phi} = \sum_{s \in \mathcal{A}} n_s^{c,\phi}. \tag{5}$$

Therefore, three different sets of counters are used, one for each state. Moreover, only the counters associated with the chosen state are updated. It is worth noting that, in order to be able to operate, this model does not require the knowledge of the correct reading frame. However, once a particular initial position has been chosen, the corresponding reading frame should be maintained,

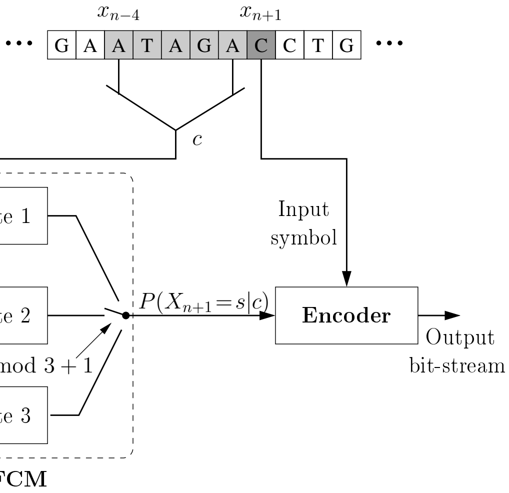


Fig. 2. Three-state model, exploiting the three-base periodicity of the DNA protein-coding regions. In this case, the probability of the next outcome, X_{n+1}, is conditioned both by the k last outcomes and by the value of (n mod $3+1$).

otherwise the statistics will become mixed and the model will not work properly. Notwithstanding, if we intend to determine the entropies associated with each of the three base positions inside the codons, we need to know which base position corresponds to each state of the model. Moreover, note that (1) needs to be modified accordingly, leading to the entropies

$$H_n^1 = -\frac{1}{\lfloor n/3 \rfloor} \sum_{i=1}^{\lfloor n/3 \rfloor} \log_2 P(X_{3i-2} = x_{3i-2}|c), \tag{6}$$

where $c = x_{3i-k-2} \ldots x_{3i-4}x_{3i-3}$,

$$H_n^2 = -\frac{1}{\lfloor n/3 \rfloor} \sum_{i=1}^{\lfloor n/3 \rfloor} \log_2 P(X_{3i-1} = x_{3i-1}|c), \tag{7}$$

where $c = x_{3i-k-1} \ldots x_{3i-3}x_{3i-2}$, and

$$H_n^3 = -\frac{1}{\lfloor n/3 \rfloor} \sum_{i=1}^{\lfloor n/3 \rfloor} \log_2 P(X_{3i} = x_{3i}|c), \tag{8}$$

where $c = x_{3i-k} \ldots x_{3i-2}x_{3i-1}$.

For the cases reported in this paper we always started the model at the beginning of a codon, implying that state one corresponds to the first base position of the codon, state two to the second base position and state three to the third base position.

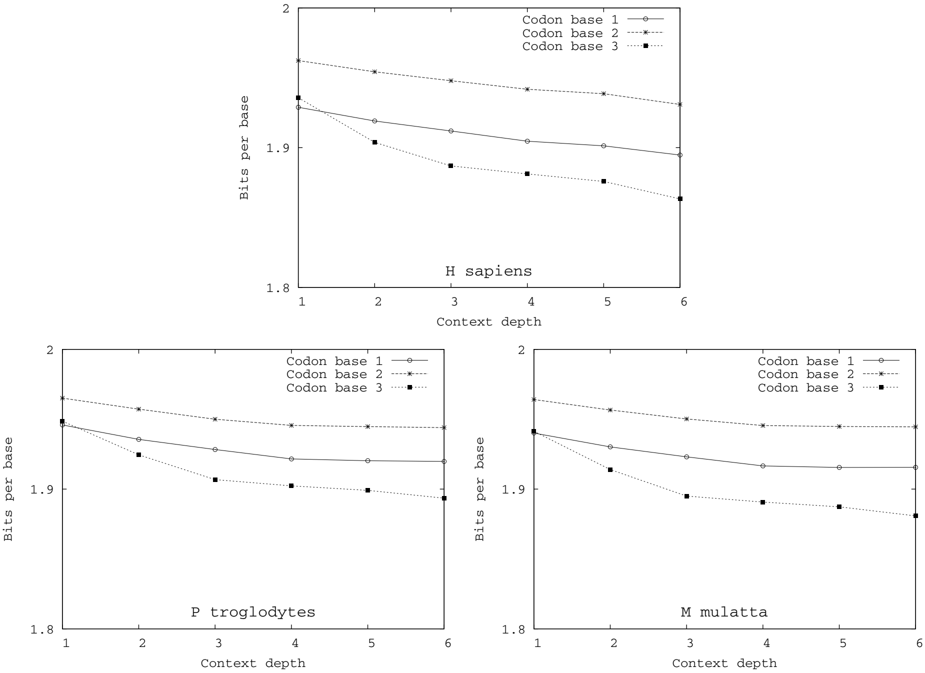


Fig. 3. Plots showing the distribution of the information among the three bases of the codon for *H. sapiens*, *P. troglodytes* and *M. mulatta*

3 Results

We ran the three-state finite-context model for the DNA sequences under analysis, using contexts of depths one to six, i.e., from $k = 1$ until $k = 6$. Figures 3–6 display graphics of the average number of bits per base obtained. Each graph contains three curves, one for each of the three bases of the codon, i.e., the values of H_n^1, H_n^2 and H_n^3 after having processed the whole sequence.

As can be seen, the plots shown in Fig. 3, corresponding to the *H. sapiens*, *P. troglodytes* and *M. mulatta* organisms, present a significant similarity. Moreover, for most of the values of k (the depth of the context) the entropy associated to the second base of the codon is the largest, followed by the first and third bases.

This behavior is also observed in the graphs of Fig. 4, where the *M. musculus* and *R. norvegicus* organisms are addressed. However, in this case, and in contrast to the previous one, it can be seen a clear inversion of the entropy of the first and third bases for $k = 1$.

Figure 5 displays the entropy graphs for the four plants used in this preliminary assessment, namely the *A. thaliana*, *P. trichocarpa*, *V. vinifera* and *R. communis*. For these organisms, the entropy of the first base is generally larger than that of the second one, which is larger than the entropy of the third base.

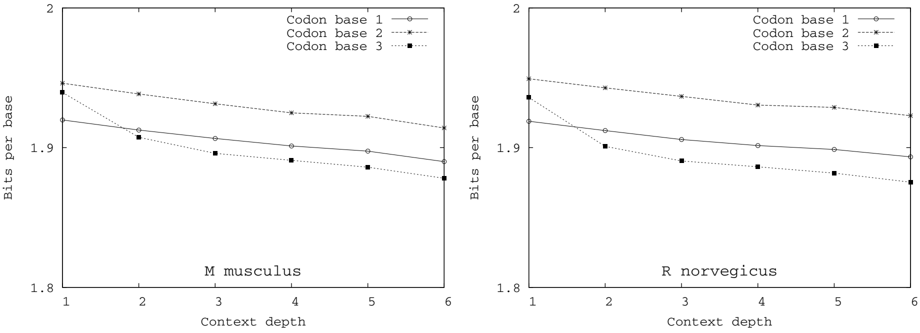


Fig. 4. Plots showing the distribution of the information among the three bases of the codon for the *M. musculus* and *R. norvegicus*

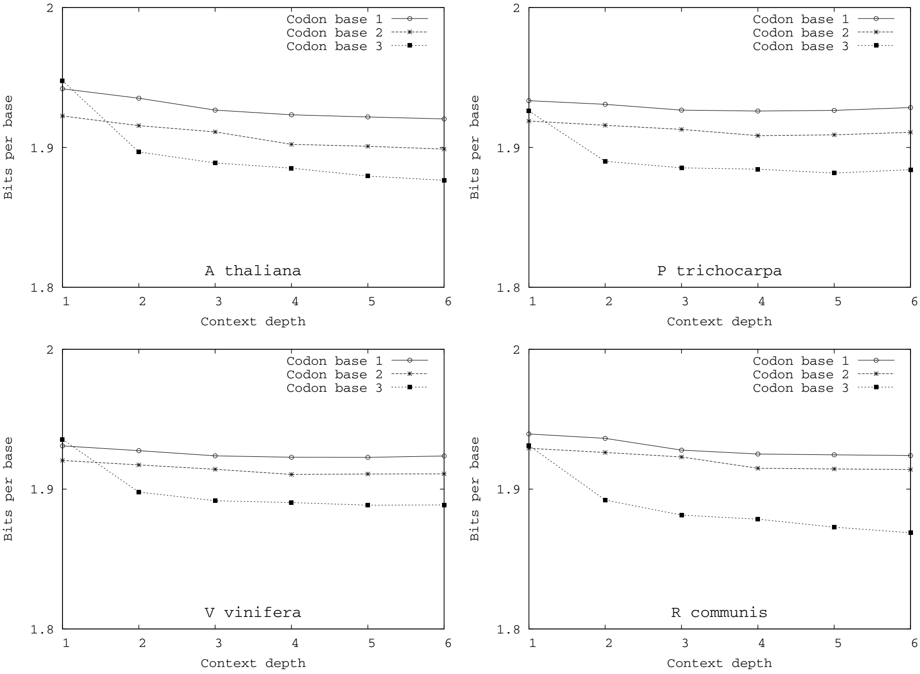


Fig. 5. Plots showing the distribution of the information among the three bases of the codon for *A. thaliana*, *P. trichocarpa*, *V. vinifera* and *R. communis*

Therefore, in comparison to the five animals, there is a change in the relative position of the curves regarding the first and second bases of the codon.

This same ordering can be found in the curves corresponding to the *S. pneumoniae*, *C. trachomatis*, *M. genitalium* and *S. mutans* organisms, presented in

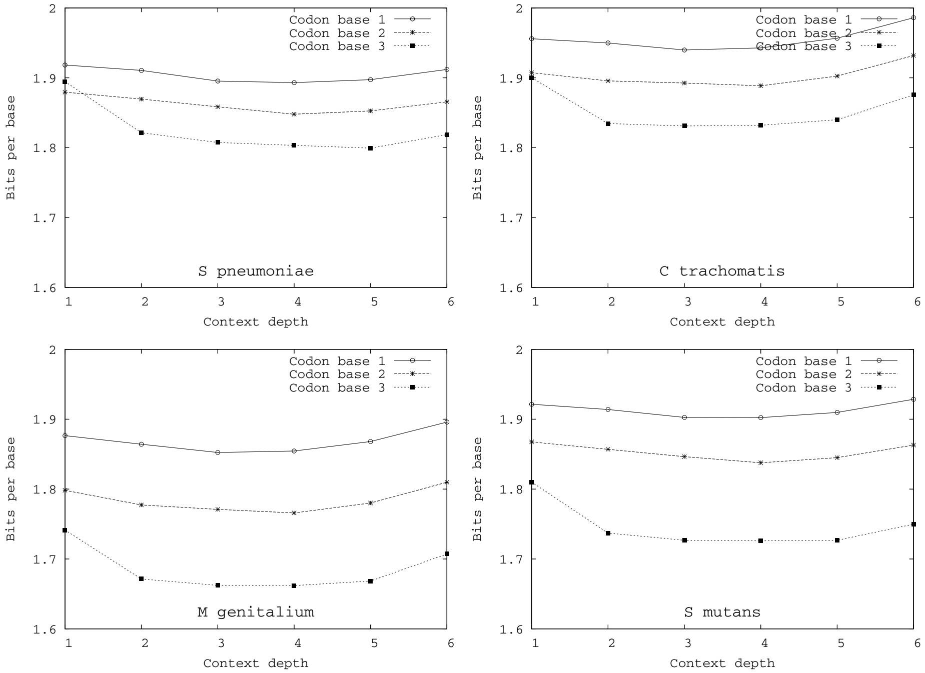


Fig. 6. Plots showing the distribution of the information among the three bases of the codon for *S. pneumoniae*, *C. trachomatis*, *M. genitalium* and *S. mutans*

Fig. 6. However, whereas for the plants the difference between the values of the upper and lower curves is typically less than 0.05 bpb, in the case of the four bacteria this difference is typically larger than 0.1 bpb.

In order to better understand the similarities and differences of the entropy values among the analyzed species, we have built a dendogram (Fig. 7) with the PHYLIP package (`http://evolution.genetics.washington.edu/phylip.html`), constructed using the unweighted pair group method with arithmetic average (UPGMA), also known as average linkage method [12]. The distance matrix was obtained by computing the Euclidean distance between vectors built from the three-state entropy vectors corresponding to context depths from one to six. Therefore, each organism is represented by a vector with eighteen elements, i.e., the concatenation of six groups of three-state entropies.

Regarding this dendogram, we have some remarks. The prokaryotes (lower branch) are correctly separated from the eukaryotes (upper branch), except for the bacterium *C. trachomatis*. Amongst the prokaryotic branch, all bacteria are correctly grouped. The clades for the animals and plants are also well identified. Amongst the plants, *P. trichocarpa* should be classified closer to *R. communis*, as they belong to the same order. As for the animals, the human should be closer to the chimpanzee, then to the Rhesus macaque, and finally to the mouse and brown rat. Tough these minor misclassifications, this methodology correctly

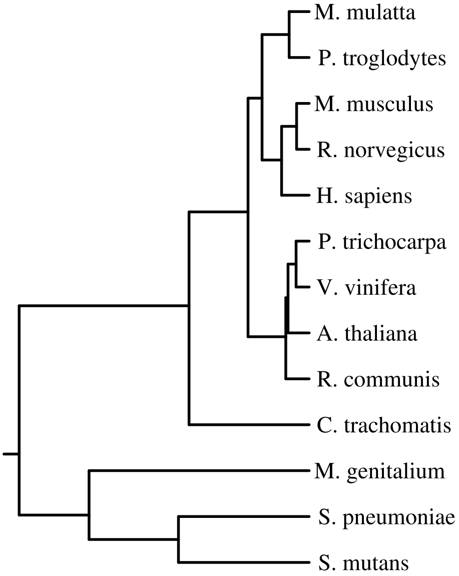


Fig. 7. Dendogram, based on UPGMA, obtained from the matrix of the Euclidean distance between the three-state entropy vectors for context depths from one to six

identifies overall clades, making these preliminary results encouraging in the exploration of three-state finite-context models for a meaningful classification of organisms.

4 Conclusion

The three-base periodicity of the exons has been used since its discovery mostly as an aid in gene finding. More recently, it was noted that the three entropy values associated to each of the three base positions of the codon are not the same, and that the differences vary from organism to organism. We refer to these three entropy values as the “three-state entropy vector” of the organism.

The work presented in this paper is a start towards a deeper investigation of the implications of this observation, particularly in what concerns its use for species classification. The preliminary results obtained suggest that the information gathered from the three-state entropy vector alone seems to be sufficient for building meaningful dendograms, encouraging further study.

Acknowledgments

This work was supported in part by the grant with the COMPETE reference FCOMP-01-0124-FEDER-007252 (FCT, Fundação para a Ciência e Tecnologia, reference PTDC/EIA/72569/2006). Sara P. Garcia acknowledges funding from the European Social Fund and the Portuguese Ministry of Science, Technology and Higher Education.

References

1. Bell, T.C., Cleary, J.G., Witten, I.H.: Text compression. Prentice-Hall, Englewood Cliffs (1990)
2. Eskesen, S.T., Eskesen, F.N., Kinghorn, B., Ruvinsky, A.: Periodicity of DNA in exons. BMC Molecular Biology 5 (2004)
3. Ferreira, P.J.S.G., Neves, A.J.R., Afreixo, V., Pinho, A.J.: Exploring three-base periodicity for DNA compression and modeling. In: Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, ICASSP-2006., Toulouse, France, vol. 5, pp. 877–880 (May 2006)
4. Fickett, J.W.: Recognition of protein coding regions in DNA sequences. Nucleic Acids Research 10(17), 5303–5318 (1982)
5. Frenkel, F.E., Korotkov, E.V.: Using triplet periodicity of nucleotide sequences for finding potential reading frame shifts in genes. DNA Research 16, 105–114 (2009)
6. Issac, B., Singh, H., Kaur, H., Raghava, G.P.S.: Locating probable genes using Fourier transform approach. Bioinformatics 18(1), 196–197 (2002)
7. Kotlar, D., Lavner, Y.: Gene prediction by spectral rotation measure: a new method for identifying protein-coding regions. Genome Research 13, 1930–1937 (2003)
8. Laplace, P.S.: Mémoire sur la probabilité des causes par les événements. Mémoires de l'Académie royale des Sciences de Paris (Savants étrangers) 6, 621–656 (1774); Reprinted in Oeuvres complètes de Laplace, Gauthier-Villars et fils, Paris, vol. 8, pp. 27–65 (1891)
9. Pinho, A.J., Neves, A.J.R., Afreixo, V., Bastos, C.A.C., Ferreira, P.J.S.G.: A three-state model for DNA protein-coding regions. IEEE Trans. on Biomedical Engineering 53(11), 2148–2155 (2006)
10. Salomon, D.: Data compression - The complete reference, 2nd edn. Springer, Heidelberg (2000)
11. Sayood, K.: Introduction to data compression, 2nd edn. Morgan Kaufmann, San Francisco (2000)
12. Sokal, R.R., Michener, C.D.: A statistical method for evaluating systematic relationships. University of Kansas Scientific Bulletin 28, 1409–1438 (1958)
13. Tiwari, S., Ramachandran, S., Bhattacharya, A., Bhattacharya, S., Ramaswamy, R.: Prediction of probable genes by Fourier analysis of genomic sequences. Bioinformatics 13, 263–270 (1997)
14. Trifonov, E.N.: 3-, 10.5, 200- and 400-base periodicities in genome sequences. Physica A 249, 511–516 (1998)
15. Yin, C., Yau, S.S.T.: Prediction of protein coding regions by the 3-base periodicity analysis of a DNA sequence. Journal of Theoretical Biology 247(1), 687–694 (2007)

A Maximum-Likelihood Formulation and EM Algorithm for the Protein Multiple Alignment Problem

Valentina Sulimova[1], Nikolay Razin[2], Vadim Mottl[3],
Ilya Muchnik[4], and Casimir Kulikowski[5]

[1] Tula State University, Lenine Ave. 92, 300600, Russia, Tula
[2] MIPT, Kerchenskaya St.1A, 117303, Russia, Moscow
[3] Computing Center of the RAS, Vavilov St.40, 119333, Russia, Moscow
[4] Rutgers University, DIMACS, New Brunswick, NJ 08901
[5] Rutgers University, Department of Computer Science, New Brunswick, NJ 08901
vsulimova@yandex.ru, nrmanutd@gmail.com, vmottl@yandex.ru,
muchnikilya@yahoo.com, kulikows@cs.rutgers.edu

Abstract. A given group of protein sequences of different lengths is considered as resulting from random transformations of independent random ancestor sequences of the same preset smaller length, each produced in accordance with an unknown common probabilistic profile. We describe the process of transformation by a Hidden Markov Model (HMM) which is a direct generalization of the PAM model for amino acids. We formulate the problem of finding the maximum likelihood probabilistic ancestor profile and demonstrate its practicality. The proposed method of solving this problem allows for obtaining simultaneously the ancestor profile and the posterior distribution of its HMM, which permits efficient determination of the most probable multiple alignment of all the sequences. Results obtained on the BAliBASE 3.0 protein alignment benchmark indicate that the proposed method is generally more accurate than popular methods of multiple alignment such as CLUSTALW, DIALIGN and ProbAlign.

Keywords: Multiple alignment problem, protein sequences analysis, EM-algorithm, HMM, common ancestor.

1 Introduction

The problem of multiple alignment of protein sequences is a fundamental problem for modern bioinformatics. It arises from applications such as secondary and tertiary structure prediction [1], reconstructing complex evolutionary histories [2, 3], locating conserved motifs and domains [4], and constructing phylogenetic trees [5].

The bioinformatics literature is replete with diverse alignment methods and tools. However, only few of them, such as multidimensional dynamic programming [6], have a mathematically strict problem formulation followed by a sound

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 171–182, 2010.

optimization procedure. Those with mathematical formulations which try to take into account information about protein evolution [7] are NP-hard and cannot be applied for aligning more than a few sequences [8]. Approximations which are not based on evolutionary trees and stars [9] and other fast heuristics, such as approaches like those which include a large family of progressive alignments [10, 11], are less biologically relevant.

Profile-based algorithms with iterative updating [12] and HMM-based approaches [13–16] have an essential common disadvantage: their results strongly depend on the initial approximation. An additional problem which is typical for HMM-based multiple alignments is that of deciding on how to select model parameters.

In this paper, we consider a new approach to the problem of multiple alignment on the basis of the simplest probabilistic model of protein evolution built as a relatively straightforward generalization of Margaret Dayhoff's PAM model (Point Accepted Mutation) developed for the alphabet of single amino acids $A = (\alpha^1 \dots \alpha^{20})$ [17]. It is assumed that the amino acid sequences $\boldsymbol{\omega_j} = (\omega_{jt} \in A, t = 1 \dots N_j)$ forming the set to be processed jointly $\Omega^* = \{\boldsymbol{\omega}_j, j = 1 \dots M\}$ are results of independent random Markov chains of insertions/substitutions applied to some unknown n-length ancestor sequences $\boldsymbol{\vartheta_j} = (\vartheta_{ji}, i = 1 \dots n), j = 1 \dots M$, specific for each $\boldsymbol{\omega}_j$ of greater length, $n \leq \min\{N_j, j = 1 \dots M\}$. The elements of the hidden sequences ϑ_{ji} are a priori assumed to be randomly and independently chosen by nature according to a sequence of n unknown probability distributions over the set of 20 amino acids $\vartheta_i \in A$.

The goal of the analysis is to estimate these probability distributions as the sought-for n-length profile playing the role of a model of the given protein set.

Such a result is not in itself a multiple alignment, but any instance of the j-th insertion/substitution transformation cuts out a n-length subsequence from the corresponding amino acid sequence $\boldsymbol{\omega}_j = (\dots \tilde{\omega}_{jt_1} \dots \tilde{\omega}_{jt_i} \dots \tilde{\omega}_{jt_n} \dots)$, which is associated with the successive elements of the supposed ancestor $(1 \dots n)$. This process will generate a vast diversity of versions of how these positions could be assembled into n relatively conserved columns.

The algorithm yields the posterior distribution over the set of possible multiple alignments relevant to the given set of proteins, covering the large variety of versions of how these positions can lead to n relatively conserved columns. So we can easily find the most probable multiple alignment.

2 Dayhoff's PAM Model of Evolution within the Amino Acid Alphabet

The formulation of the multiple alignment problem considered in the present paper is based on the pioneering model of amino acid evolution Point Accepted Mutation (PAM) introduced by M. Dayhoff in 1978 [17]. The PAM model represents predispositions of amino acids towards mutual mutative transformations as a square matrix of conditional probabilities that amino acid α^i will be substituted at the next step of evolution by amino acid α^j :

$$\boldsymbol{\Psi} = \big(\psi(\alpha^j|\alpha^i), \alpha^i, \alpha^i \in A\big)(20 \times 20), \sum_{\alpha^j \in A} \psi(\alpha^j|\alpha^i) = 1. \tag{1}$$

The main probabilistic assumption underlying the PAM model is that the Markov chain defined by the transition matrix $\boldsymbol{\Psi}$ possesses the two classical properties:

- ergodicity, namely, existence of a final probability distribution over the set of amino acids $\xi(\alpha^j) = \sum_{\alpha^i \in A} \xi(\alpha^i)\psi(\alpha^j|\alpha^i)$,
- and reversibility $\xi(\alpha^i)\psi(\alpha^j|\alpha^i) = \xi(\alpha^j)\psi(\alpha^i|\alpha^j)$.

3 Model of the Common Origin of a Set of Proteins

Let Ω be the set of all finite amino acid sequences $\boldsymbol{\omega} = (\omega_t, t = 1, \ldots, N)$, $\omega_t \in A = \{\alpha^1, \ldots, \alpha^{20}\}$. We shall use also the notation $\Omega_n = \{\boldsymbol{\omega} = (\omega_t, t = 1, \ldots, N), \omega_t \in A, N = n\}$ for the set of all sequences having a fixed length n.

We proceed from the following probabilistic assumptions on the common origin of the proteins to be analyzed jointly $\Omega^* = \{\boldsymbol{\omega}_j, N_j \geq n, j = 1, \ldots, M\}$. These assumptions are essentially based on those taken in [18], aimed at an evolution-based pairwise comparison of proteins. On the one hand, we simplify them, because we use here only one particular class of described in [18] random transformations of sequences. But, on the other hand, we generalize this model because several amino acid sequences can be jointly processed here instead of just two.

Hypothesis 1. *Each of the amino acid sequences in the given set $\Omega^* = \{\boldsymbol{\omega}_j = (\omega_{jt}, t = 1, \ldots, N_j), j = 1, \ldots, M\}$ is considered as having evolved from its specific hidden ancestor $\boldsymbol{\vartheta}_j = (\vartheta_{ji} \in A, i = 1, \ldots, n) \in \Omega_n$ through independent known random transformations represented by the family of conditional probability distributions $\varphi_{jn}(\boldsymbol{\omega}|\boldsymbol{\vartheta}_j)$, $\sum_{\boldsymbol{\omega} \in \Omega_{N_j}} \varphi_{jn}(\boldsymbol{\omega}|\boldsymbol{\vartheta}_j) = 1$.*

Hypothesis 2. *Let the length n of the random ancestors $\boldsymbol{\vartheta}_j \in \Omega_n$ be fixed, and their elements ϑ_{ji} be drawn from the alphabet of amino acids in accordance with a common sequence of unknown independent probability distributions $\big(\beta_i(\vartheta), \vartheta \in A\big)$, $\sum_{\vartheta \in A} \beta_i(\vartheta) = 1$.*

Each of these distributions is completely represented by a 20-dimensional vector of probabilities $\boldsymbol{\beta}_i = (\beta_i^1, \ldots, \beta_i^{20}) \in \mathbb{R}^{20}$, $\sum_{k=1}^{20} \beta_i^k = 1$. It should be noticed that the sequence of distributions $\bar{\boldsymbol{\beta}} = (\boldsymbol{\beta}_i, i = 1, \ldots, n) \in \mathbb{R}^{20n}$ corresponds to the notion of the probabilistic profile, which is commonly adopted in bioinformatics.

This profile is the common parameter of identical independent probability distributions of the hidden ancestors $\boldsymbol{\vartheta}_j$ for each of the observed amino acid sequences :

$$p_n(\boldsymbol{\vartheta}_j|\bar{\boldsymbol{\beta}}) = p_n(\vartheta_{j1}, \dots, \vartheta_{jn}|\boldsymbol{\beta}_1, \dots, \boldsymbol{\beta}_n) = \prod_{i=1}^{n} \beta_i(\vartheta_{ji}). \tag{2}$$

So, it is assumed here that the entire given set of amino acid sequences $\Omega^* = \{\boldsymbol{\omega}_j, N_j \geq n, j = 1, \dots, M\}$ has evolved from the same hidden profile $\bar{\boldsymbol{\beta}}$.

Hypothesis 3. *The transformation $\varphi_{Nn}(\boldsymbol{\omega}|\boldsymbol{\vartheta})$ of the n-length ancestor $\boldsymbol{\vartheta}_j \in \Omega_n$ into some random protein $\boldsymbol{\omega}_j$ of random length $N_j \geq n$ is a concatenation of the two following random mechanisms.*

The first step of the transformation is a random choice of the structures $\boldsymbol{v} = (1 \leq v_1 \leq \dots \leq v_n)$ of transformations independently for each of the resulting sequences $\boldsymbol{\vartheta} \to \boldsymbol{\omega}$, $v_n \leq N$, namely, assigning the positions $\boldsymbol{\omega} = (\dots \bar{\omega}_{v_1} \dots \bar{\omega}_{v_i} \dots \bar{\omega}_{v_n} \dots)$ into which the elements of the ancestor $\boldsymbol{\vartheta} = (\vartheta_1, \dots, \vartheta_n)$ will be mapped. These positions are called in [18] *key positions.* The apriori distributions of the respective key-position vectors $q_{Nn}(\boldsymbol{v}) = q_{Nn}(v_1, \dots, v_n)$ are assumed to take into account only the gaps between the key positions $v_i - v_{i-1}$ and be indifferent to the lengths of both tails v_1 and $N - v_n$. Distributions $q_{Nn}(\boldsymbol{v})$ are necessarily specific for each of the lengths $N_j, j = 1, \dots, M$, because of the constraints $v_n \leq N_j$:

$$\begin{aligned} &q_{N_j n}(\boldsymbol{v}|a,b) = \begin{cases} \propto \prod_{i=2}^{n} g(v_i - v_{i-1}|a,b), v_n \leq N_j, \\ = 0, v_n > N_j, \end{cases} \\ &g(v_i - v_{i-1}|a,b) \propto \begin{cases} 1, d_i = v_i - v_{i-1} = 1, \\ \exp\left[-c(a + b(v_i - v_{i-1}))\right], d_i > 1, \end{cases} \\ &a > 0, b > 0, c > 0. \end{aligned} \tag{3}$$

Such a distribution ranks one long gap as more preferable than several short ones adding up to the same length.

The second step is filling the key positions in the resulting sequences with random amino acids in accordance with Dayhoff's conditional mutation probabilities $\psi(\omega_{v_i}|\vartheta_i)$ ([1]). The structure-dependent conditional transformation distributions are assumed to be completely uniform relative to amino acids in other positions:

$$\eta_n(\boldsymbol{\omega}|\boldsymbol{\vartheta}, \boldsymbol{v}) \propto \prod_{i=1}^{n} \psi(\omega_{v_i}|\vartheta_i), \tag{4}$$

where $v \in \mathbb{V}_{Nn}$ for each specific $N = N_j$, and $\mathbb{V}_{Nn}$ is the set of all n-length transformation structures with respect to the length of the sequence $1 \leq v_1 < \dots < v_n \leq N$.

It follows from Hypotheses 3 that each transformation $\boldsymbol{\vartheta} \to \boldsymbol{\omega} = \boldsymbol{\omega}_j, N = N_j$, is defined as the mixture

$$\varphi_{Nn}(\boldsymbol{\omega}|\boldsymbol{\vartheta}) = \sum_{v \in \mathbb{V}_{Nn}} q_{Nn}(\boldsymbol{v})\eta_n(\boldsymbol{\omega}|\boldsymbol{\vartheta}, \boldsymbol{v}), \boldsymbol{\omega} \in \Omega_N, \tag{5}$$

and, in accordance with Hypotheses 2, the marginal conditional distribution of the sequence of length N is expressed as

$$f_N(\boldsymbol{\omega}|\bar{\boldsymbol{\beta}}) = \sum_{v \in \mathbb{V}_{Nn}} q_{Nn}(\boldsymbol{v})\zeta_n(\boldsymbol{\omega}|\bar{\boldsymbol{\beta}}, \boldsymbol{v}), \boldsymbol{\omega} \in \Omega_N, \tag{6}$$

where

$$\zeta_n(\boldsymbol{\omega}|\bar{\boldsymbol{\beta}}, \boldsymbol{v}) = \sum_{\boldsymbol{\vartheta} \in \Omega_n} \eta_n(\boldsymbol{\omega}|\boldsymbol{\vartheta}, \boldsymbol{v}) p_n(\boldsymbol{\vartheta}|\bar{\boldsymbol{\beta}}) \tag{7}$$

is the conditional distribution of a single random sequence with respect to the assumed structure $\boldsymbol{v} \in \mathbb{V}_{Nn}$ of its evolving from the unknown random ancestor of length n.

4 Maximum-Likelihood Estimation of the Common Profile

It follows from Hypothesis 1 that the joint distribution of independent sequences making the given set $\Omega^* = \{\boldsymbol{\omega}_j, j = 1 \dots M\}$ is the product of individual distributions (6)

$$F(\Omega^*|\bar{\boldsymbol{\beta}}) = \prod_{j=1}^{M} f_{N_j}(\boldsymbol{\omega}_j|\bar{\boldsymbol{\beta}}). \tag{8}$$

This is, in effect, a likelihood function with respect to the sought-for profile whose maximum-likelihood estimate will be given by the maximum point of this function:

$$\hat{\bar{\boldsymbol{\beta}}} = \arg\max_{\bar{\boldsymbol{\beta}}} \ln F(\Omega^*|\bar{\boldsymbol{\beta}}) = \arg\max_{\bar{\boldsymbol{\beta}}} \sum_{j=1}^{M} \ln \sum_{v \in \mathbb{V}_{N_j n}} q_{N_j n}(\boldsymbol{v})\zeta_n(\boldsymbol{\omega}_j|\bar{\boldsymbol{\beta}}, \boldsymbol{v}). \tag{9}$$

The presence of a sum within the logarithm seems to hinder the maximization. But on the other hand, the set of sequences $\Omega^* = \{\boldsymbol{\omega}_j, j = 1 \dots M\}$ is the observable part of the two-component random object (Ω^*, Υ_n) whose hidden part $\Upsilon_n = (\boldsymbol{v}_j \in \mathbb{V}_{N_j n}, j = 1 \dots M)$ is the collection of the sequence-specific transformation structures.

This fact suggests the application of the Expectation-Maximization (EM) principle, which results, in this case, in the following iterative procedure $s = 1, 2, 3, \dots$, starting with an initial approximation $\bar{\boldsymbol{\beta}}_0 = (\boldsymbol{\beta}_{1,0}, \dots, \boldsymbol{\beta}_{n,0}) \subseteq \mathbb{R}^{20n}$.

Let $\bar{\boldsymbol{\beta}}_s = (\boldsymbol{\beta}_{1,s}, \dots, \boldsymbol{\beta}_{n,s})$ be approximation at step s, and

$$p_{it}(\bar{\boldsymbol{\beta}}_s, \boldsymbol{\omega}_j) = P(v_{ij} = t|\bar{\boldsymbol{\beta}}_s, \boldsymbol{\omega}_j) \tag{10}$$

be the a posteriori probability of the event $v_{ij} = t$ in the transformation structure $\boldsymbol{v}_j = (1 \leq v_{j1} < \cdots < v_{jn})$, which means that the i-th element $\boldsymbol{\beta}_{i,s}$ of the profile $\bar{\boldsymbol{\beta}}_s = (\boldsymbol{\beta}_{1,s}, \ldots, \boldsymbol{\beta}_{n,s})$ is associated with the t-th element ω_{jt} of the j-th sequence $\boldsymbol{\omega}_j = (\omega_{j1}, \ldots, \omega_{jN_j})$. The next value of the i-th element of the profile $\boldsymbol{\beta}_{i,s+1} = (\beta^1_{i,s+1}, \ldots, \beta^{20}_{i,s+1}) \in \mathbb{R}^{20}$ is defined as

$$\begin{cases} (\beta^1_{i,s+1}, \ldots, \beta^{20}_{i,s+1}) = \underset{(\beta^1_i, \ldots, \beta^{20}_i) \in \mathbb{R}^{20}}{\arg\max} \sum\limits_{l=1}^{20} h^l_i \ln \sum\limits_{k=1}^{20} \psi(\alpha^l|\alpha^k)\beta^k_i, \\ \sum\limits_{k=1}^{n} \beta^k_i = 1, \, \beta^k_i \geq 0, \, k = 1, \ldots, 20, \end{cases} \tag{11}$$

where $h^l_i = \sum\limits_{j=1}^{M} \sum\limits_{t=1}^{N_j} I[\omega_{jt} = \alpha^l] p_{it}(\bar{\boldsymbol{\beta}}_s, \boldsymbol{\omega}_j)$, and indicator function $I[\omega_{jt} = \alpha^l] = 1$ if the condition $\omega_{jt} = \alpha^l$ is met, or 0 if not. Solving this problem is provided by the well-known gradient projection algorithm.

Theorem 1. *The choice of $\bar{\boldsymbol{\beta}}_{s+1} = (\boldsymbol{\beta}_{1,s+1} \ldots \boldsymbol{\beta}_{n,s+1})$ in accordance with (11) provides that the inequality $F(\Omega^*|\bar{\boldsymbol{\beta}}_{s+1}) > F(\Omega^*|\bar{\boldsymbol{\beta}}_s)$ holds true at each step s while $\nabla_{\bar{\boldsymbol{\beta}}} F(\Omega^*|\bar{\boldsymbol{\beta}}_s) \neq \mathbf{0}$; if $\nabla_{\bar{\boldsymbol{\beta}}} F(\Omega^*|\bar{\boldsymbol{\beta}}_s) = \mathbf{0}$ then $F(\Omega^*|\bar{\boldsymbol{\beta}}_{s+1}) = F(\Omega^*|\bar{\boldsymbol{\beta}}_s)$.*

Proof. The proof directly follows from the standard derivation and reasoning for EM algorithms.

Computation of posterior probabilities (10) is also a standard problem, in this case, in the theory of hidden Markov models, because the random transformation structure $\boldsymbol{v} = (1 \leq v_1 < \cdots < v_n)$ with independent gaps defined by (3) is a Markov process for each amino acid sequence in the data set under analysis $\Omega^* = \{\boldsymbol{\omega}_j, j = 1, \ldots, M\}$.

5 Choosing Main Parameters of the Algorithm

The main parameters of the proposed algorithm are the length n of the common profile $\bar{\boldsymbol{\beta}} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_n)$ and the initial approximation for the profile $\bar{\boldsymbol{\beta}}_0 = (\boldsymbol{\beta}_{0,1}, \ldots, \boldsymbol{\beta}_{0,n})$.

These parameters can be chosen by a number of different ways. For example it appears reasonable to take the value n which provides the minimum average entropy of the profile columns:

$$\hat{n} = \arg\min_n \Big(-\frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{n} \beta^k_i \ln \beta^k_i \Big). \tag{12}$$

This criterion satisfies the requirement of the final goal of the analysis, which is understood as finding the most conserved columns of amino acids in the given set of proteins.

When the likelihood function (8) has only one maximum, i.e., the set of its stationary points $\{\bar{\boldsymbol{\beta}} : \nabla_{\bar{\boldsymbol{\beta}}} F(\Omega^*|\bar{\boldsymbol{\beta}}) = \mathbf{0}\} \subseteq \mathbb{R}^{20n}$ is convex, the choice of the initial approximation $\bar{\boldsymbol{\beta}}_0 = (\boldsymbol{\beta}_{0,1}, \ldots, \boldsymbol{\beta}_{0,n})$ is not too significant. For instance, it is enough to take the sequence of uniform distributions over the set of amino acids $\boldsymbol{\beta}_{0,i} = (1/20, ..., 1/20) \in \mathbb{R}^{20}$, $i = 1, \ldots, n$.

However, when the sequences under analysis $\Omega^* = \{\boldsymbol{\omega}_j, j = 1 \ldots M\}$ have low identity, the likelihood function has a tendency to be not unimodal. In this paper, we choose both parameters n and $\bar{\boldsymbol{\beta}}_0 \in \mathbb{R}^{20n}$ at once by computing them using the multiple alignment obtained by some different method, for example ProbAlign. The number of columns without gaps in this alignment defines the length of the common profile n, and the distributions of amino acids in these columns are taken as the initial distributions $\boldsymbol{\beta}_{0,1}, \ldots, \boldsymbol{\beta}_{0,n}$. The efficiency of such approach is confirmed by results of experiments.

6 The Most Probable Multiple Alignment

The n-column profile $\hat{\hat{\boldsymbol{\beta}}}$ found as the maximum-likelihood estimate (9) of the fuzzy common subsequence of the assumed preset length n in the given set of proteins may be considered as the goal of their joint analysis. But the final a posteriori probabilities $p_{it}(\hat{\hat{\boldsymbol{\beta}}}, \boldsymbol{\omega}_j) = P(v_{ij} = t | \hat{\hat{\boldsymbol{\beta}}}, \boldsymbol{\omega}_j)$ (10) of the positions associated with each of the single amino acid sequences for successive elements of the supposed common ancestor $(1, \ldots, n)$ show a vast variety of versions of how these positions could be assembled into relatively conserved columns. This is the posterior distribution over the set of possible multiple alignments relevant to the given set of proteins.

The a posteriori most probable one will be given by the solutions of separate optimization problems corresponding to single proteins $\boldsymbol{\omega}_j, j = 1 \ldots M$:

$$\begin{cases} \boldsymbol{v}_j = \underset{v_1, \ldots, v_n}{\arg\max} \prod_{i=1}^{n} p_{iv_i}(\hat{\hat{\boldsymbol{\beta}}}, \boldsymbol{\omega}_j), \\ v_{ji} \geq v_{j,i-1}, i = 2 \ldots n. \end{cases} \tag{13}$$

This is a standard dynamic programming problem.

7 Experimental Results and Discussion

7.1 Characteristic Features of the Proposed Alignment and Its Visual Representation

It should be noted, that the form of multiple alignment obtained in accordance with (13) is different from the most conventional form of multiple alignment. The proposed approach actually produces only n columns without gaps, each of which corresponds to the respective i-th ($i = 1 \ldots n$) element of the alleged common ancestor of the sequences. Other amino acids are not aligned. An example of a visual representation of a multiple alignment produced in accordance with our approach is presented in Figure 1,b. In contrast, Figure 1,a shows the traditional form of the benchmark multiple alignment produced by biologists.

The main part of our alignment in Figure 1,b is separated from the rest at the left and at the right by three empty columns, each of which contains only gaps. Left fragments of the sequences, which precede the main part, are flushed right, whereas right fragments, following the main part, are flushed left. Amino acids located between the ungapped aligned columns are conventionally flushed to the centers of idle intervals.

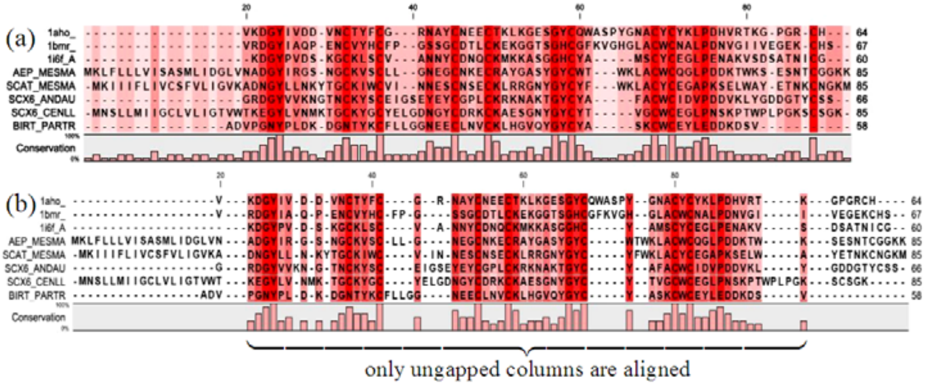


Fig. 1. Examples of multiple alignments: (a) manually-refined benchmark alignment and (b) alignment produced by the proposed approach

7.2 Alignment Benchmark

We tested our approach on a subset of BAliBASE 3.0 [20], which is the database of manually-refined multiple sequence alignments specifically designed for the evaluation and comparison of multiple sequence alignment programs.

For our tests we used families of short proteins from 3 different sets of BALiBase RV11, RV12 and RV20. The set RV11 contains equidistant families with sequence identity less than 20%, while RV12 contains equidistant families with sequence identity between 20% and 40%. Both of these sets lack sequences with large internal insertions (> 35 residues). The set RV20 contains families with > 40% similarity and an orphan sequence which shares less than 20% similarity with the rest of the family.

The main characteristics of the tested families are presented in Table 1.

7.3 Determining Prediction Accuracy

Given a true and an estimated multiple sequence alignment, the accuracy of the estimated alignment is usually computed using two measures: the sum-of-pairs (SP) and the true column (TC) scores. The SP score is a measure of the number of correctly aligned residue pairs divided by the number of aligned residue pairs in the true alignment, and TC is the number of correctly aligned columns divided by the number of columns in the true alignment. Both of them are standard measures of computing alignment accuracy. The source code of a

Table 1. Characteristics of the considered families

Set	Family name	Description	Number of sequences	Lengths	Number of columns without gaps in benchmark
RV11	1aab	high mobility group protein	4	83 – 91	76
	1aboA	SH3	8	52 – 193	47
	1bbt3	foot-and-mouth disease virus	6	186 – 283	150
	1csy	SH2	4	104 – 540	91
	1dox	ferredoxin [2fe-2s]	4	97 – 337	78
RV12	1axo	toxin II	8	58 – 85	51
	1fj1A	homeodomain	9	49 – 254	49
	1hfh	factor h	4	118 – 129	115
	1hpi	high-potential iron-sulfur protein	6	71 – 85	65
	1krn	serine protease	5	79 – 475	78
RV20	1idy	myb dna-binding domain	38	54 – 256	45
	1pamA	cyclodextrin	16	247 – 527	215
	1pgtA	glutathione	31	202 – 244	175
	1tvxA	pertussis toxin	29	64 – 167	50
	1ubi	ubiquitin	47	76 – 155	67

program for computing these scores is available for download at the BALiBase site [20]. However, this program is not accurate enough, it has a tendency to overstate the TC and SP scores and, moreover, to give values greater then 1, which is impossible given the definition of these scores.

In this connection, we use our implementation of the procedure for computing Bali-scores. It should be noticed that our procedure, in contrast to the original one, takes into account only pairs of amino acids which belong to the columns without gaps. This approach is much more appropriate for the principle of multiple alignment proposed in this paper but, as a rule, yields smaller values of scores.

7.4 Experimental Setup and Results

For each family under consideration, four multiple alignments were computed. Three of them were produced by the popular multiple alignment tools CLUSTALW, DI-ALIGN and ProbAlign, which were run on their respective servers. The value of the constant for the ProbAlign algorithm, called ”the thermodynamic temperature”, was chosen to be 5 as the most reasonable value according to publications [14]. The remaining parameters of this and other algorithms were set at their default values.

Finally, the 4-th alignment was produced in accordance with the proposed approach, started from the resulting alignment of ProbAlign as initial approximation.

The four-way comparison of SP and TC scores is presented in Table 2. The best values of scores are highlighted in bold font.

Table 2. Results of comparing multiple alignment procedures. TC/SP scores of multiple alignments produced by different algorithms.

Set	Family	CLUSTALW	DIALIGN	ProbAlign	The proposed approach
RV11	1aab	0.92/0.96	0.91/0.93	0.83/0.87	**0.99/0.99**
	1aboA	0.00/0.38	0.00/0.00	0.00/**0.54**	0.00/0.45
	1bbt3	0.00/0.20	0.00/0.00	**0.29/0.42**	0.28/0.36
	1csy	0.37/0.42	0.31/0.37	0.46/0.56	**0.51/0.56**
	1dox	0.00/0.24	0.40/0.46	0.62/0.71	**0.64/0.75**
RV12	1axo	0.29/0.54	0.54/0.64	0.69/0.87	**0.87/0.93**
	1fj1A	**1.00/1.00**	0.69/0.76	0.79/0.84	**1.00/1.00**
	1hfh	0.68/0.78	0.39/0.53	**0.78/0.85**	0.75/**0.85**
	1hpi	0.59/0.72	0.37/0.57	0.40/0.55	**0.75/0.82**
	1krn	0.53/0.69	0.47/0.68	0.60/0.75	**0.79/0.88**
RV20	1idy	0.00/**0.62**	0.00/0.00	0.00/0.33	0.00/0.60
	1pamA	0.43/0.77	0.29/0.58	**0.74/0.84**	0.69/0.83
	1pgtA	**0.47**/0.49	0.14/0.52	0.26/**0.69**	0.27/0.68
	1tvxA	0.00/**0.64**	0.00/0.00	0.00/0.41	0.00/0.46
	1ubi	0.00/**0.68**	0.00/0.03	**0.09**/0.49	0.08/0.48
	mean	0.35/0.61	0.30/0.41	0.44/0.65	**0.51/0.71**

As can be seen, in more than half of all the above cases our proposed approach yields the best results. The greatest success is achieved for families of the set RV12. But also for other families, the TC and SP scores of our approach are larger, in many cases, than scores of the main competitor ProbAlign. As a result, the average scores for the proposed approach are the best.

In addition, some interesting statistics computed from Table 2 are presented in Table 3 for comparing the proposed approach with the ProbAlign.

Table 3. Statistics computed from Table 2 for comparing the proposed approach with the ProbAlign

	TC / SP
The number of cases when our proposed approach is better or equal	11(73%) / 10(67%)
The mean increment of scores	0.112 / 0.127
The mean percentage increment of scores	23% / 21%
The mean decrement of scores	0.025 / 0.036
The mean percentage decrement of scores	6% / 7.1%

8 Conclusions

In this paper we have proposed and tested a new formulation of the multiple alignment problem. It is based on a deliberately simplified model of proteins evolution, which is a direct generalization of the PAM model for amino acids. For solving the respective optimization problem we have used an iterative procedure based on the EM-algorithm.

The first experiments show that the proposed approach outperforms other methods of multiple alignment by mean values of TC and SP scores. It does not yield the best scores for all considered cases, but it can be seen that, as a rule, our method shows small decreasing and large increasing of scores in contrast to other methods.

References

1. Rost, B., Sander, C., Schneider, R.P.: - an automatic server for protein secondary structure prediction. Computational Applications in Biosciences 10, 53–60 (1994)
2. Notredame, C.: Recent progresses in multiple sequence alignment: a survey. Pharmacogenomics 3(1), 131–144 (2002)
3. Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids, p. 356. Cambridge University Press, Cambridge (1998)
4. Attwood, T.K.: The PRINTS database: A resource for identification of protein families. Brief Bioinformatics 3, 252–263 (2002)
5. Saitou, N., Nei, M.: The neighbor-joining method: A new method for reconstructing phylo-genetic trees. Molecular Biology 212, 403–428 (1987)
6. Sankoff, D., Cedergren, R.J.: Simultaneous comparison of three or more sequences related by a tree. In: Sankoff, D., Kruskal, J.B. (eds.) Time Warps, String Edits and Macromolecules: The Theory and Practice of Sequence Comparison, pp. 253–263. Addison-Wesley, Reading (1989)
7. Altschul, S.F., Lipman, D.J.: Trees, stars, and multiple biological sequence alignment. SIAM J. Appl. Math. 49, 197–209 (1989)
8. Todd Wareham, H.: A simplified proof of the NP- and MAX SNP-hardness of multiple sequence tree alignments. J. Comput. Biol. 2(4), 509–514 (1995)
9. Carrillo, H., Lipman, D.: The multiple sequence alignment problem in biology. SIAM J. Appl. Math. 48, 1073–1082 (1988)
10. Notredame, C., Higgins, D.G., T-Coffee, H.J.: A novel method for fast and accurate multiple sequence alignment. J. Mol. Biol. 302, 205–217 (2000)
11. Subramanian, A.R., Kaufmann, M., Morgenstern, B.: DIALIGN-TX: Greedy and progres-sive approaches for segment-based multiple sequence alignment. Algorithms for Molecular Biology 3, 6 (2008)
12. Barton, G.J., Sternberg, M.J.E.: A strategy for the rapid multiple alignment of protein se-quences. J. Mol. Biol. 198, 327–337 (1987)
13. Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: Biological sequence analysis: Probabilistic models of proteins and nucleic acids. Cambridge University Press, Cambridge (1998)
14. Roshan, U., Libesay, D.R.: Probalign: Multiple Sequence Alignment Using Partition Function Posterior Probabilities. Oxford University Press, Oxford (2005)
15. Do, C.B., Mahabhashyam, M.S., Brudno, M., Batzoglou, S.: ProbCons: Probabilistic Consis-tency-based Multiple Sequence Alignment. Genome Res. 15, 330–340 (2005)
16. Pei, J., Grishin, N.V.: PROMALS: Towards accurate multiple sequence alignments of dis-tantly related proteins. Bioinformatics 23, 802–808 (2007)

17. Dayhoff, M.O., Schwarts, R.M., Orcutt, B.C.: A model of evolutionary change in proteins. Atlas of Protein Sequences and Structures 5(suppl. 3), 345–352 (1978)
18. Sulimova, V., Mottl, V., Mirkin, B., Muchnik, I., Kulikowski, C.: A class of evolution-based kernels for protein homology analysis: A generalization of the PAM model. In: Proceedings of the 5th International Symposium on Bioinformatics Research and Applications, May 13-16, pp. 284–296. Nova Southeastern University, Ft. Lauderdale (2009)
19. Thompson, J.D., Koehl, P., Ripp, R., Poch, O.: BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. Proteins 61, 127–136 (2005)
20. BALiBASE3.0: A benchmark alignment database home page, `http://www-bio3d-igbmc.u-strasbg.fr/~julie/balibase/index.html`

Polynomial Supertree Methods Revisited

Malte Brinkmeyer, Thasso Griebel, and Sebastian Böcker

Department of Computer Science, Friedrich Schiller University, 07743 Jena, Germany
{malte.b,thasso.griebel,sebastian.boecker}@uni-jena.de

Abstract. Supertree methods allow to reconstruct large phylogenetic trees by combining smaller trees with overlapping leaf sets, into one, more comprehensive supertree. The most commonly used supertree method, matrix representation with parsimony (MRP), produces accurate supertrees but is rather slow due to the underlying hard optimization problem. In this paper, we present an extensive simulation study comparing the performance of MRP and the polynomial supertree methods *MinCut Supertree*, *Modified MinCut Supertree*, *Build-with-distances*, *PhySIC*, and *PhySIC_IST*. We consider both quality and resolution of the reconstructed supertrees. Our findings illustrate the trade-off between accuracy and running time in supertree construction, as well as the pros and cons of voting- and veto-based supertree approaches.

1 Introduction

In recent years, supertree methods have become a familiar tool for building large phylogenetic trees. Supertree approaches combine input trees with overlapping taxa sets into one large and more comprehensive tree. Since the introduction of the term *supertree* and the first formal supertree method [1], there has been a continuous development of supertree methods, see e.g. [2]. The supertree approach has certain advantages over standard phylogenetic reconstruction methods, both on the theoretical and practical side [3]: It allows to combine heterogeneous data sources, such as DNA hybridization data, morphological data, and protein sequences. Furthermore, it enables inference for groups where most species are represented by very few genes and sequences, and the major part of sequences is available only for few species, which makes deriving a balanced molecular phylogeny difficult. On the theoretical side, it is well known that inferring optimal trees from sequences is a computationally hard problem under the maximum likelihood (ML) [4] and the maximum parsimony (MP) criterion [5], so we have to rely on heuristics that cannot guarantee to find the optimal solution. Even for a moderate number of species, the sheer size of tree space prohibits to search for optimal trees under these criteria. Current supertree methods can roughly be subdivided into two major families: matrix representation (MR) and polynomial, mostly graph-based methods. The former encode the inner vertices of all input trees as partial binary characters in a matrix, which is analyzed using an optimization or agreement criterion to yield the supertree. Matrix representation with parsimony (MRP) [6,7], the first matrix-based method,

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 183–194, 2010.

is still by far the most widely used supertree method today. Other variants have been proposed using different optimization criteria, e.g. matrix representation with flipping (MRF) [8] or matrix representation with compatibility (MRC) [9]. All MR methods have in common that the underlying optimization problems are computationally hard, and heuristic search strategies have to be used. As for ML and MP, it is unclear how close the resulting tree is to the optimal one.

Graph-based methods make use of a graph to encode the topological information given by the input trees. This graph is used as a guiding structure to build the supertree top-down from the root to the leaves. The *MinCut Supertree* algorithm (MC) [10] and a modified version, *Modified MinCut Supertree* (MMC) [11], use a minimum-cut approach to construct a supertree if the input trees are conflicting. The *Build-with-distances* algorithm (BWD) [12] is the first graph-based method that uses branch length information from the input trees to build the supertree. Ranwez et al. [13] presented a new graph-based method, the *PhySIC* algorithm. The method ensures that the reconstructed supertree satisfies two properties: it contains no clade that directly or indirectly contradicts the input trees and each clade in the supertree is present in an input tree, or is collectively induced by several input trees. Supertree methods guaranteeing the first property are called *veto* methods, that, in case of highly conflicting and/or poorly overlapping input trees, tend to produce unresolved supertrees. Scornavacca et al. [14] presented a modified version of *PhySIC*, *PhySIC_IST*, that tries to overcome this drawback by proposing non-plenary supertrees (i.e. supertrees that do not necessarily contain all taxa from the input trees), while still assuring the properties mentioned above. *PhySIC_IST* works in a stepwise fashion, iteratively adding leaves to a starting tree consisting of two nodes. In contrast to MR methods, the MC, MMC, BWD, *PhySIC* and *PhySIC_IST* algorithms have polynomial running time.

As an increasing number of supertree methods is available, simulation studies are needed to compare the behavior and performance of the methods under various conditions. The advantage of simulation studies is that the results of different methods can be compared to a known model tree and thus the methods can be compared at an absolute scale. Although several simulation studies focusing on different aspects of the investigated supertree have been carried out (e.g. [15], [16]), they have only just begun to provide useful comparisons of alternative methods. This paper focuses a special subset of supertree construction methods: we are in particular interested in the comparison of the accuracy of the MRP method as exponent of the MR based family of supertree methods, for which it has been shown that they are accurate and highly resolved but require long running times, and the mentioned polynomial supertree methods, which are swift but possibly less accurate and in case of *PhySIC* and *PhySIC_IST*, also possibly less resolved. Here, we present a large-scale simulation study conducted to compare the accuracy and the resolution of MRP, MC, MMC, BWD, *PhySIC*, and *PhySIC_IST* supertrees. Additionally, we explore new variations of BWD, trying to improve its performance. Our simulation study follows the established general scheme to assess the performance of supertree methods: (1) Construction of a model tree under a Yule process, (2) simulation of DNA alignments along that tree, (3) random deletion of a proportion of taxa (4) reconstruction of trees by ML, (5) construction of supertree from the inferred ML trees, and, finally (6) comparison of the supertree to the model tree using distance and similarity measures and evaluation of its resolution. Our results demonstrate that the BWD and the *PhySIC_IST* method perform significantly better than MC and MMC, and are, with respect to the accuracy of the reconstructed supertree, sometimes even comparable with MRP. Moreover, as we also consider the resolution of the supertrees, our findings illuminate the trade-off between accuracy and running time in supertree construction, as well as the pros and cons of voting and veto approaches.

2 Methods under Consideration

Build and MinCut supertrees. The first graph-based supertree method is the *Build* algorithm [17], an all-or-nothing approach that encodes the input trees into a graph structure and returns a supertree only if the input trees are compatible. The *MinCut Supertree* algorithm (MC) [10] was the first extension of *Build* capable of returning a supertree if the input trees are not compatible. The incompatibilities are resolved by deleting a minimal amount of information present in the input trees in order to allow the algorithm to proceed. Page [11] presented a modified version of MC that uses more information from the input trees. By using a variation of the underlying graph structure, the *Modified MinCut Supertree* (MMC) algorithm ensures to incorporate all clades from the input trees with which no single tree directly disagrees.

Build-with-distances supertrees. Willson [12] presented another extension of *Build*, the *Build-with-distances* (BWD) algorithm that, in addition to the branching information in the input trees, uses branch lengths to build the supertree. Basically, the method follows the same recursive schema as *Build*, MC, and MMC. The main observation underlying the BWD algorithm is that branch lengths may carry phylogenetic information, such as an estimated number of mutations. Clearly, the use of branch length is only justified if these are comparable amongst the input trees, i.e. the input to the method has to be carefully selected, or the branch lengths have to be reconciled or normalized in some way. The BWD algorithm incorporates branch lengths from the input trees to add more information to the used graph. BWD uses different *support functions*, which basically estimate the evidence that two taxa should be in the same clade of the supertree. We find that in our simulation study using the *accumulated confirmed support function* (SAC) consistently outperforms other support functions. Hence, we will concentrate on SAC in our evaluations as well as a new established support function, SACmax. Details are deferred to the full version of this paper. In contrast to the minimum-cut approach used by MC and MMC, Willson uses the *bisection method* to deal with incompatible input trees.

PhySIC and PhySIC_IST supertrees. Unlike all methods mentioned before, the *PhySIC* algorithm [13] applies a *veto* philosophy. Following Ranwez et al. [13], supertree methods are either *voting* or *veto* procedures. A characteristic of the voting approach is that the input trees are asked to vote for clades in the phylogeny to be inferred; the most frequent alternatives are chosen. Voting methods resolve conflicts by using an optimization criterion in order to select between different possible topologies [18]. When input trees conflict, voting methods as MRP can infer supertrees in which clades are present that are contradicted by each of the input trees (e.g. [19]). In contrast to voting methods, the veto approach is more conservative in handling conflicts among the input trees: the inferred supertree has to respect the phylogenetic information of each source tree and is not allowed to contain any clade that is contradicted by one or more of the input trees. Thus, conflicts among the input trees are removed [18], for example by proposing multifurcations in the supertree or by pruning rogue taxa. Scornavacca et al. [14] presented *PhySIC_IST*, a modification of the *PhySIC* algorithm, aiming to circumvent a main drawback of veto supertree methods: These tend to return highly unresolved supertrees if the input trees imply a high degree of incompatibility, or do not have a high degree of overlap. To overcome this shortcoming, *PhySIC_IST* modifies the original approach non-plenary supertrees (i.e. supertrees that do not necessarily contain all taxa present in the input trees) and by using a preprocessing step called STC (Source tree correction), which analyzes and modifies the input trees concerning the conflicts they contain. Basically, it removes parts of each source tree that significantly conflict with other source trees.

Matrix Representation with Parsimony (MRP). MRP encodes the inner vertices of all input trees as partial binary characters in a matrix, which is analyzed using the parsimony criterion as objective function. Two different coding schemes have been suggested to decompose trees into an matrix representation: the Baum-Ragan (BR) and the Purvis (PU) coding scheme. Furthermore, two kinds of parsimony can be used: reversible Fitch parsimony and irreversible Camin-Sokal parsimony. MRP with BR and Fitch is commonly used and generally accepted as standard method for supertree construction.

3 Simulation Study

In this section we present a large scale simulation study conducted to evaluate the accuracy and resolution of the methods MRP, MC, MMC, *PhySIC*, *PhySIC_IST*, and BWD (with modifications). An overview of the simulation layout can be found in Figure 1. Each step is described in detail below.

Generating Model Trees and DNA Sequences. We generated model trees according to a stochastic Yule birth process using the default parameters of the YULE_C procedure from the program r8s [20] with either 48, 96 and 144 taxa. For each model tree size we generated 100 different model tree replicates. By the use of the program Seq-gen v1.3.2 [21], nucleotide sequences were simulated

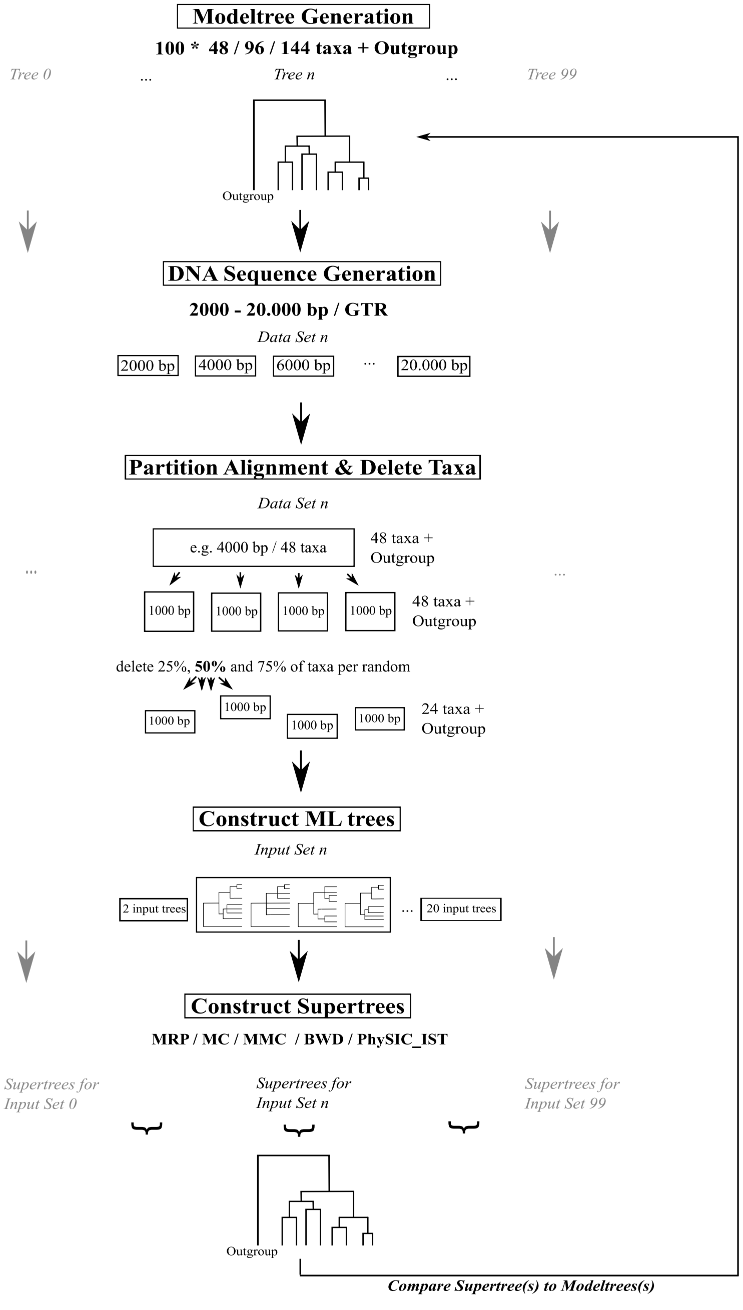


Fig. 1. Simulation pipeline overview

along each of the model trees according to the general time reversible process (GTR) model [22] with parameters Lset Base = (0.3468 0.3594 0.0805), Rmat = (0.6750 27.9597 1.1677 0.4547 20.8760), gamma rate heterogeneity $\alpha = 1.1999$ and PINVAR = 0.4954, taken from [23]. For each model tree we generated sequences ranging from 2000 to 20000 base pairs in steps of 2000, yielding in ten different sequence alignments per model tree.

Generating Input Trees. All models of molecular substitution implemented in Seq-Gen assume evolution is independent and identical at each site. Hence, contiguous blocks of sequences represent randomly subdivided data set. We partitioned each alignment into blocks of 1000-base pair data sets and randomly deleted 25%, 50% and 75% of sequences from each alignment to simulate different taxa overlaps observed in real data sets. For each resulting alignment block we inferred a maximum likelihood tree using RAxML v 7.0.0. [24] with default parameters. This yields in sets ranging from 2 to 20 input trees belonging to one model tree.

Supertree construction. MRP supertrees were estimated using PAUP* 4.0b10 [25] with TBR branch swapping as heuristic search, random addition of sequences and a maximum 10.000 trees in memory. The search time for a single MRP supertree run was delimited by 300 seconds. The strict consensus tree of all most-parsimonious trees was used as final MRP tree. We computed MC as well as the BWD supertrees using our own implementations embedded in our software framework EPoS[1]. MMC trees were generated using Rod Page's implementation[2]. For the *PhySIC* and *PhySIC_IST* supertrees we used the implementations provided from the authors of the corresponding papers[3][4]. To test a broader range of the *PhySIC_IST* STC preprocess (-c option), we used 0, 0.5 and 1 as parameters. In our setting, the results for 0 and 0.5 are similar; therefore, only the 0 and 1 parameter results are shown. In the following we will refer these as *PhySIC IST 0* and *PhySIC IST 1.*

Measuring accuracy and resolution. To evaluate the accuracy of the supertrees build by the different methods we compared the supertrees to the model trees using different distance and similarity scores, namely the Robinson-Foulds metric (*RF distance*) [26], the maximum agreement subtree score, *MAST score* [27], and the *triplet distance* [11]. We stress that each of these methods has its particular shortcomings, for a discussion and implementation details see the full version of this paper. The resolution was measured as the number of clades in the inferred supertree relative to the total number of clades on a fully binary tree of the same size (n - 2 for an unrooted tree, where n = number of taxa). Resolution varies between 0 and 1, where 0 indicates a unresolved bush and 1 indicates a complete binary supertree.

[1] http://bio.informatik.uni-jena.de/epos/

[2] http://darwin.zoology.gla.ac.uk/~rpage/supertree/

[3] http://www.atgc-montpellier.fr/physic/binaries.php

[4] http://www.atgc-montpellier.fr/physic_ist/

4 Results

Results of our simulation for 48 taxa are reported in Figure 2, where we plot resolution and triplet distance against the number of input trees. In Figure 3, we use our simulations on 96 taxa and plot MAST score and RF distance against number of input trees. One would expect that results improve if more input data becomes available, as this helps us to identify bogus information. Hence, triplet distance and RF distance should decrease, whereas the MAST score should increase when more input trees are available to the supertree method. We now discuss the observed patterns in more detail.

Resolution. In our setting *PhySIC* mostly returns star trees. The two variations of the BWD algorithm build the most resolved supertrees compared to all other methods, independent from the deletion frequency the number of input trees. In general, this also holds for MMC and MC. In case of 25% deletion frequency, MRP behaves similar to MMC and MC, but is significantly less resolved than all others at 75% deletion frequency. In case of 25% and 50% deletion frequency *PhySIC_IST* 0 produces more resolved supertrees than *PhySIC_IST* 1. In comparison to all methods, the *PhySIC_IST* 1 supertrees are least resolved. With 75% deletion frequency, the resolutions of the *PhySIC_IST* 0 and *PhySIC_IST* 1 supertrees are quite similar. In general, one can see that BWD as an advanced graph-based supertree method outperforms the classical parsimony approach (MRP) as well as the conservative, veto based algorithm (*PhySIC_IST*) in terms of resolution. The results also clearly show that the more conservative *PhySIC_IST* 1 produces less resolved trees than *PhySIC_IST* 0, reflecting the influence of the STC parameter.

Triplet Distance. In the majority of cases, MC algorithm performs worst compared to all other algorithms and an increasing number input of trees has no positive effect on the accuracy. The MMC algorithm generally performs better than MC, but its accuracy also does not significantly increase with the number of input trees, except for the case of 25% deletion frequency. Both BWD methods perform better than MC/MMC but their accuracy also does not significantly benefit from a growing number of input trees. In case of 25% and 50% deletion frequency, *PhySIC_IST* 1 produces less accurate supertrees with an increasing number of input trees. This can be explained by the decreasing resolution, which has direct impact on the number of matching triplets. In contrast, the accuracy of *PhySIC_IST* 0 is relatively stable and independent of the deletion frequency and the number of input trees. MRP always performs better than the algorithms mentioned so far. The number of input trees has in general a slight positive effect on the accuracy.

MAST score. In general, the MC algorithm provides supertrees with the worst MAST score compared to all other methods. Only in the case of 25% deletion frequency MC performs slightly better than *PhySIC_IST* 1. *PhySIC_IST* 1 behaves generally like the MC algorithm. *PhySIC_IST* 0 produces supertrees with

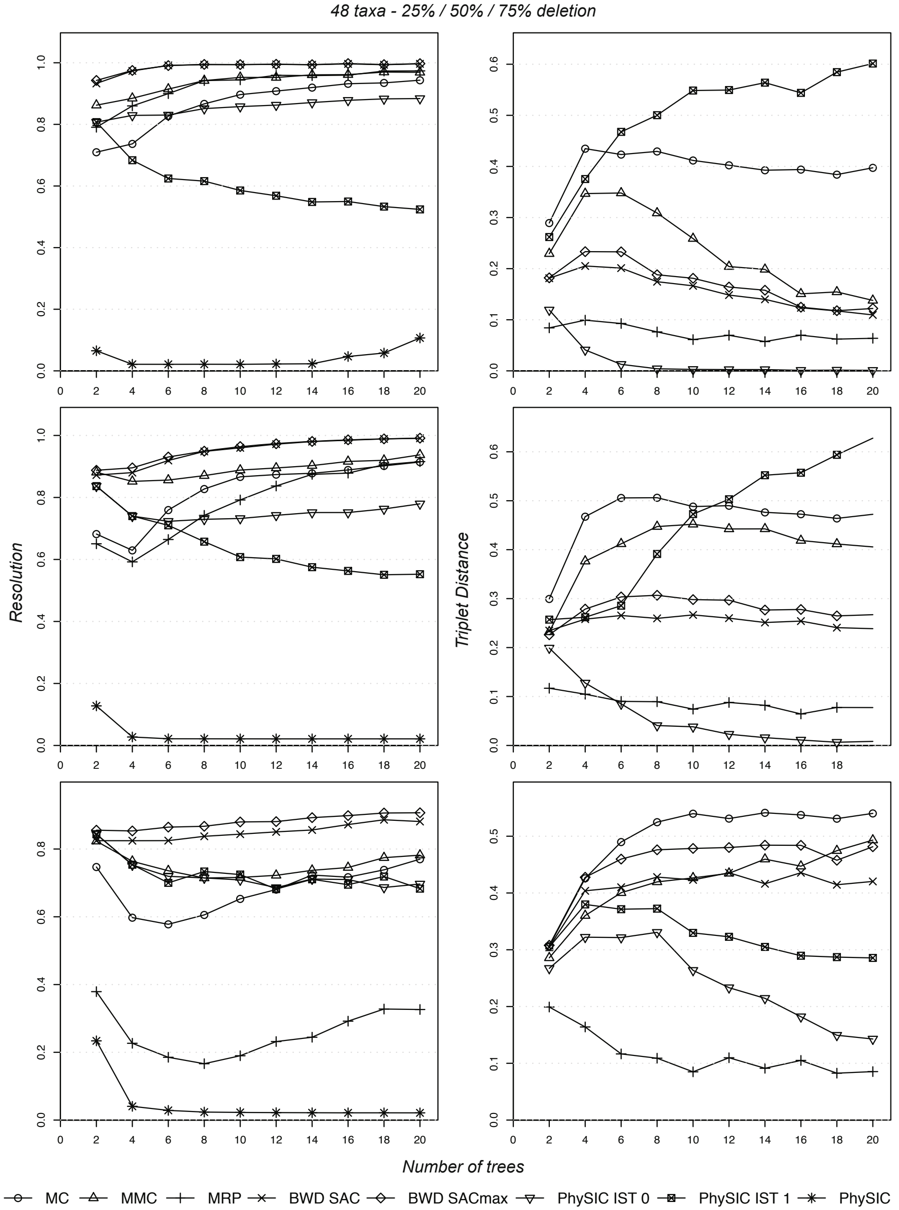


Fig. 2. The left column of the figure shows the average resolution of the supertrees constructed from model trees with 48 taxa and different taxon deletion rates (top 25%, middle 50%, bottom 75%). The right column shows the average triplet distances of the supertrees constructed from model trees with 48 taxa and different taxon deletion rates (top 25%, middle 50%, bottom 75%).

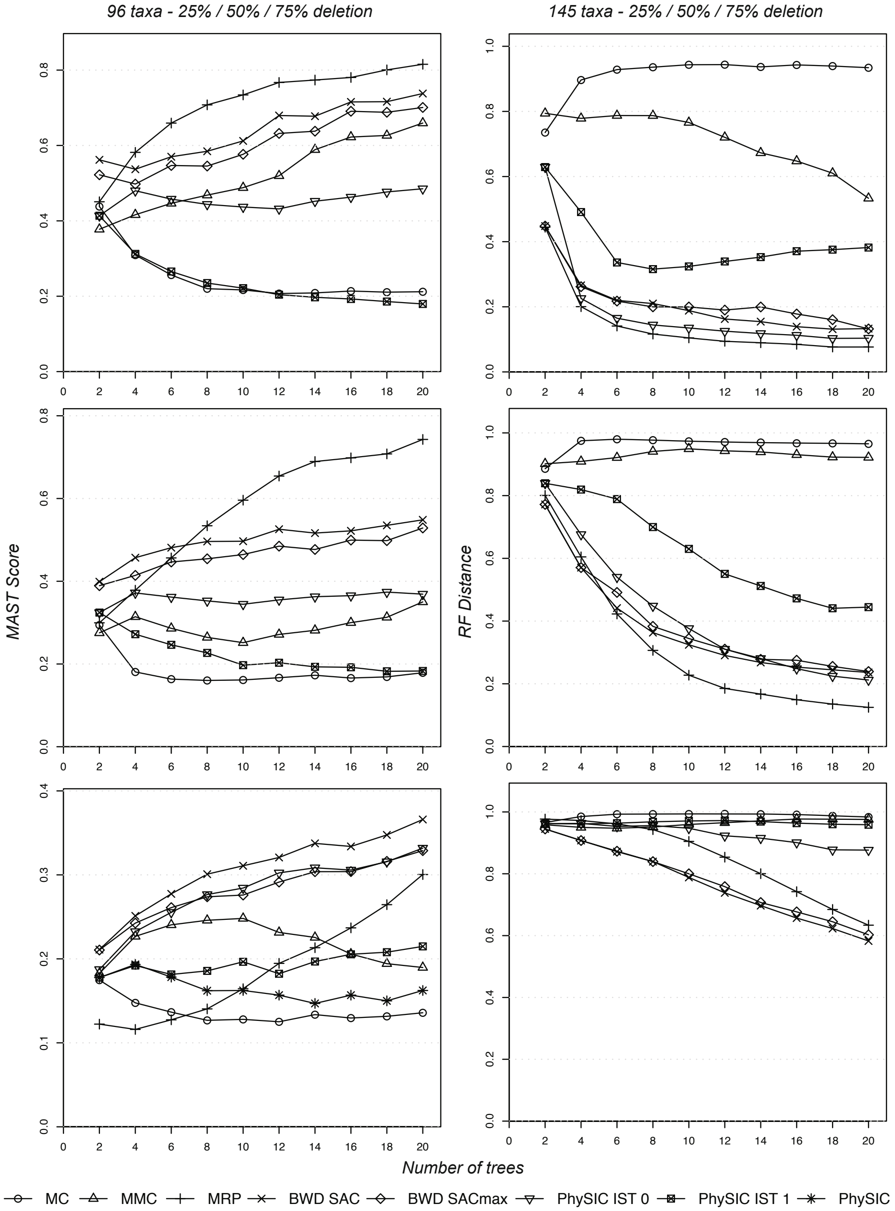


Fig. 3. The left column of the figure shows the average MAST scores of the supertrees constructed from model trees with 96 taxa and different taxon deletion rates (top 25%, middle 50%, bottom 75%). The right column shows the average RF-Distance of the supertrees constructed from model trees with 145 taxa. Note that the MAST values are similarity scores and RF values are distances.

a considerably better MAST scores than MC and *PhySIC_IST* 1, but the number of input trees has no significant effect on the MAST score. MMC algorithm performs slightly better than *PhySIC_IST* 0 and 1 as well as the MC method in case of 25% deletion frequency With 25% deletion frequency MMC's MAST score increases with more input, in both other cases the score is relatively constant. The MRP method performs better than all other methods in the case of 25% and 50% deletion frequency and significantly benefits from a growing number of source trees. With 75% deletion frequency the MAST score of all methods under consideration are quite low and MRP can only outperform *PhySIC_IST* 1, *PhySIC_IST* 0, MC and MMC with a large number of input trees. For 75% deletion frequency, the BWD methods outperform MRP and show an increasing MAST score with an increasing number of input trees. With 25% and 50% deletion frequency, both BWD methods are only outperformed by MRP. In both cases the number of input trees has a positive effect on the MAST score.

RF distance. For all combinations of model tree sizes and deletion probabilities, the MC methods performs worst compared to all other methods. As with the triplet distance and the MAST score, MMC shows an improvement over the original method. The *PhySIC_IST* 1 performs generally better than MC and MMC. The number of input trees has in general a positive effect on the RF distance. In case of 25% and 50% deletion frequency all other methods perform similar, although MRP produces slightly better results.

5 Conclusion

We have presented a large-scale simulation study to assess and compare the accuracy and the resolution of polynomial supertree methods and the *de facto* standard supertree method MRP. Our results show that recent, polynomial supertree methods can sometimes compete with the classical MRP approach while providing a significantly better running time (which did not exceed a few seconds for all polynomial methods). The BWD method that incorporates branch length information from the input trees, significantly enhances the graph-based approaches concerning accuracy and resolution, without sacrificing short running times. For example, the MAST score at 75% deletion (Fig. 3 left) is consistently better for BWD than for MRP, for any number of input trees. Veto approach such as *PhySIC* have certain appealing properties but also certain drawbacks: the resolution of reconstructed supertree rapidly decreases when there are too many conflicts among input trees, and/or small taxon overlap. *PhySIC_IST*, in combination with the STC preprocessing, significantly enhances the veto approach in terms of resolution and accuracy, but at the cost that taxa are not included in the supertree.

For medium-sized studies with hundreds of taxa and tens of trees, we propose to use several of the supertree methods presented here, and to manually compare the results. But when the sheer size of the problem renders it impossible to use matrix-representation methods such as MRP, then novel polynomial-time methods such as BWD and *PhySIC_IST* will greatly improve the quality of results, compared to early methods such as MC or MMC. Although formal supertree methods have been around for a quarter of a century, our simulation also show that there is still much room for improvement, and that novel ideas and methods can greatly improve the quality of constructed supertree.

References

1. Gordon, A.D.: Consensus supertrees: The synthesis of rooted trees containing overlapping sets of labelled leaves. J. Classif. 3, 335–348 (1986)
2. Bininda-Emonds, O.R.P. (ed.): Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life. Computational Biology Book Series, vol. 4. Kluwer Academic, Dordrecht (2004)
3. Bininda-Emonds, O.R.P.: Supertree construction in the genomic age. Methods Enzymol. 395, 745–757 (2005)
4. Roch, S.: A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. IEEE/ACM Trans. Comput. Biol. Bioinform. 3(1), 92–94 (2006)
5. Foulds, L.R., Graham, R.L.: The Steiner problem in phylogeny is NP-complete. Adv. Appl. Math. 3, 43–49 (1982)
6. Baum, B.R.: Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees. Taxon 41(1), 3–10 (1992)
7. Ragan, M.A.: Matrix representation in reconstructing phylogenetic relationships among the eukaryotes. Biosystems 28(1-3), 47–55 (1992)
8. Chen, D., Eulenstein, O., Fernández-Baca, D., Sanderson, M.: Minimum-flip supertrees: complexity and algorithms. IEEE/ACM Trans. Comput. Biol. Bioinform. 3(2), 165–173 (2006)
9. Ross, H.A., Rodrigo, A.G.: An assessment of matrix representation with compatibility in supertree construction. In: Bininda-Emonds, O.R.P. (ed.) Phylogenetic Supertrees (combining information to reveal the tree of life), vol. 3, pp. 35–63. Kluwer Academic Publishers, Dordrecht (2004)
10. Semple, C., Steel, M.: A supertree method for rooted trees. Discrete Appl. Math. 105(1-3), 147–158 (2000)
11. Page, R.D.M.: Modified mincut supertrees. In: Guigó, R., Gusfield, D. (eds.) WABI 2002. LNCS, vol. 2452, pp. 537–552. Springer, Heidelberg (2002)
12. Willson, S.J.: Constructing rooted supertrees using distances. Bull. Math. Biol. 66(6), 1755–1783 (2004)
13. Ranwez, V., Berry, V., Criscuolo, A., Fabre, P.-H., Guillemot, S., Scornavacca, C., Douzery, E.J.P.: PhySIC: a veto supertree method with desirable properties. Syst. Biol. 56(5), 798–817 (2007)
14. Scornavacca, C., Berry, V., Lefort, V., Douzery, E.J.P., Ranwez, V.: PhySIC_IST: cleaning source trees to infer more informative supertrees. BMC Bioinformatics 9, 413 (2008)
15. Bininda-Emonds, O.R.P., Sanderson, M.J.: Assessment of the accuracy of matrix representation with parsimony analysis supertree construction. Syst. Biol. 50(4), 565–579 (2001)
16. Levasseur, C., Lapointe, F.-J.: Total evidence, average consensus and matrix representation with parsimony: What a difference distances make. Evol. Bioinform. 2, 249–253 (2006)
17. Aho, A.V., Sagiv, Y., Szymanski, T.G., Ullman, J.D.: Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. SIAM J. Comput. 10(3), 405–421 (1981)

18. Thorley, J.L., Wilkinson, M.: A view of supertree methods. In: Jannowitz, M.F., Lapointe, F.J., McMorris, F.R., Roberts, F.S. (eds.) Bioconsensus, vol. 61. The American Mathematical Society, Providence (2003)
19. Goloboff, P.A., Pol, D.: Semi-strict supertrees. Cladistics 18(5), 514–525 (2002)
20. Sanderson, M.J.: r8s: inferring absolute rates of molecular evolution and divergence times in the absence of a molecular clock. Bioinformatics 19(2), 301–302 (2003)
21. Rambaut, A., Grassly, N.C.: Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. Comput. Appl. Biosci. 13(3), 235–238 (1997)
22. Yang, Z.: Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. J. Mol. Evol. 39(3), 306–314 (1994)
23. Higdon, J.W., Bininda-Emonds, O.P., Beck, R.M.D., Ferguson, S.H.: Phylogeny and divergence of the pinnipeds (carnivora: Mammalia) assessed using a multigene dataset. BMC Evol. Biol. 7, 216 (2007)
24. Stamatakis, A.: RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics 22(21), 2688–2690 (2006)
25. Swafford, D.: Paup*: Phylogenetic analysis using parsimony (*and other methods), Version 4 (2002)
26. Robinson, D.F., Foulds, L.R.: Comparison of phylogenetic trees. Math. Biosci. 53(1-2), 131–147 (1981)
27. Gordon, A.D.: On the assessment and comparison of classifications. In: Tomassine, R. (ed.) Analyse de Données et Informatique, Le Chesnay, INRIA, France, pp. 149–160 (1980)

Enhancing Graph Database Indexing by Suffix Tree Structure

Vincenzo Bonnici[1], Alfredo Ferro[1], Rosalba Giugno[1],
Alfredo Pulvirenti[1], and Dennis Shasha[2]

[1] Dipartimento di Matematica ed Informatica, Università di Catania, Catania, Italy
vincenzo.bonnici@hotmail.it, ferro@dmi.unict.it,
giugno@dmi.unict.it, apulvirenti@dmi.unict.it
[2] Courant Institute of Mathematical Sciences, New York University, New York, USA
shasha@cs.nyu.edu

Abstract. Biomedical and chemical databases are large and rapidly growing in size. Graphs naturally model such kinds of data. To fully exploit the wealth of information in these graph databases, scientists require systems that search for all occurrences of a query graph. To deal efficiently with graph searching, advanced methods for indexing, representation and matching of graphs have been proposed.

This paper presents GraphGrepSX. The system implements efficient graph searching algorithms together with an advanced filtering technique.

GraphGrepSX is compared with SING, GraphFind, CTree and GCoding. Experiments show that GraphGrepSX outperforms the compared systems on a very large collection of molecular data. In particular, it reduces the size and the time for the construction of large database index and outperforms the most popular systems.

Keywords: subgraph isomorphism, graph database search, indexing, suffix tree, molecular database.

1 Introduction and Related Work

Application domains such as bioinformatics and cheminformatics represent data as graphs where nodes are basic elements (i.e. proteins, atoms, etc.) and edges model relations among them. In these domains, graph searching plays a key role. For example, in computational biology locating subgraphs matching a specific topology is useful to find motifs of networks that may have functional relevance. In drug discovery, the main task is to find novel bioactive molecules, i.e., chemical compounds that, for example, protect human cells against a virus. One way to support the solution of this task is to analyze a database of known and tested molecules with the aim of building a classifier which predicts whether a novel molecule will be active or not. Future chemical tests can focus on the most promising candidates (see Fig. 1).

The graph searching problem can be formalized as follows. Given a database of graphs $D = \{G_1, G_2, \ldots, G_n\}$ (e.g. collection of molecules, etc.) and a query

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 195–203, 2010.

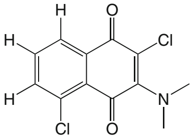
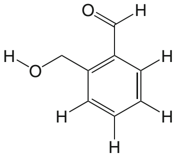
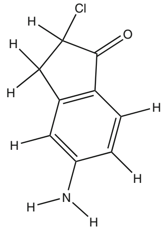
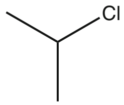
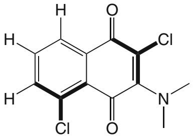
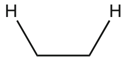
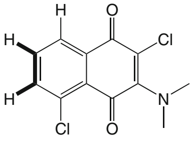
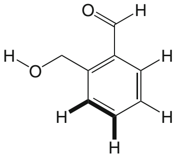
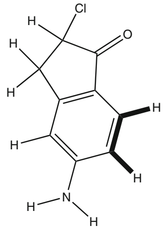
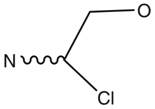
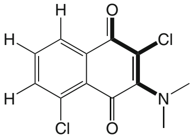
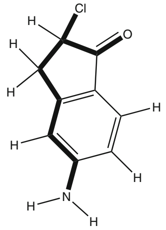
Graph Database	G_1	G_2	G_3
			
Exact Query	Number of Occurrences in Graphs		
			
Q_1	2		
			
Q_2	2	3	1
Approximate Query	Graphs Containing the Query		
			
Q_3	1		1

Fig. 1. Querying a database of graphs. Graphs represent molecules. During the match process, edge information is ignored. Query occurrences are shown in bold. For Q_2, since query matches overlap, only one occurrence in each molecule is depicted. The number of occurrences is also given. Molecular descriptions include hydrogen atoms for search accuracy. In a context where hydrogen atoms are not considered, query Q_2 is present 11 times in G_1, 6 in G_2 and 10 in G_3. The approximate query specifies any path of an unspecified length between atoms C and N. Approximate queries may also contain atoms with unknown label (they match any atom). In this paper we do not exploit approximate queries since the compared systems do not deal with such scenarios.

graph Q (e.g pattern), find all graphs in D containing Q as a subgraph. Ideally, all occurrences of Q in those graphs should be detected. Since most of these problems involve solutions of the graph isomorphism problem, an efficient exact solution cannot exist. In order to make searching time acceptable, research efforts have tried to reduce the search space by filtering out the graphs that do not contain the query. After candidate graphs have been selected, an exhaustive search on these graphs must be performed. This step is implemented either by traditional (sub)graph-to-graph matching techniques [7,3] or by an implementation that extends the SQL algebra [8].

For a database of graphs a filter limits the search to only possible candidate graphs. The idea is to extract structural features of graphs and store them in a global index. When a query graph is presented, its own structural features are extracted and compared with the features stored in the index to check compatibility [4,12,10]. Most existing systems use subgraphs (paths [4,12,5,6], trees [15,1], graphs [14]) of small size (typically not larger than 10 nodes).

In order to apply such systems on large graphs, SING [5] tool stores the starting node of each feature. This is done to capture the notion of features that are branches of trees. The matching algorithm is also modified to start the search on a selected node whose label is present in the query and not from a random one.

However, even though small subgraphs are used, the size of the index and its time construction may be high. Therefore, high-support/high-confidence mining rules are used to index only frequent and non-redundant subgraphs (i.e. a subgraph is redundant when its presence in a graph can be predicted by the presence of its subgraphs) [15,1,14]. More precisely, gIndex [14] stores, in a compact tree, all discriminat and frequent subgraphs. FGIndex [14] uses two indexes: the first one is stored in main memory, the second one is on disk. In order to assign a feature to an index, the query is performed on the main-memory-resident index. If it doesn't return any result, this index is used to identify the blocks of the secondary memory index to be loaded. GraphFind [6] uses the low-support data mining technique (Min-Hashing [2]) to reduce the index size. It is shown that such a mining technique can be successfully applied to enhance other systems such as gIndex. The above tools all require an effective but expensive data mining step.

Several indexes are based on capturing other discrimant characteristics of the graph. CTree [9] applies a graph closure to the database graphs, aligning vertices and edges using a fast approximate algorithm called neighbor biased mapping. It stores an ouput synthesized graph in a R-tree-like data structure. During the filtering phase an approximate match is executed on the closure graphs of the tree in a top down approach. Ctree spends much of its time in this matching phase. GCoding [16] uses graph signatures made by concatenating vertex signatures. A vertex signature is built from its label, neighbor labels and higher eigenvalues of the adjacency matrix of a tree representing all length n paths starting from a random node. The signature graph set is inserted into a B-tree-like structure index. In this way GCoding allows a compact representation of the indexes, but the cost of the eigenvalue computation and the high number of produced candidates reduce the method's efficiency.

In this paper, we propose GraphGrepSX, a novel approach inspired by the GraphGrep ([12,6]) system. GraphGrepSX uses paths of bounded length as features stored in a Suffix tree [13] structure. By exploiting path prefix sharing, the algorithm reduces redundancies and achieves a more compact representation of the index. This approach is particularly effective on graphs with a small label space (e.g. chemical molecules). In such a case, the same partial combination of labels could be present several times in the features of different graphs. Although such a representation is very natural and simple, GraphGrepSX is able to speed up both the index construction and the filtering phases. Moreover since it has a low index loading time, it is suitable for searching on dynamic datasets. To evaluate the performance of GraphGrepSX, we compare it with the most prominent graph search systems.

2 GraphGrepSX

GraphGrepSX uses paths of bounded length as features stored in a Suffix tree [13] structure. In what follows we describe the phases of the method.

2.1 Preprocessing Phase

The preprocessing phase extracts the features from the graph database and inserts them into the global index. Every node v_j of a graph G_i of the database is visited by a depth-first search. During this phase, all the paths of length up to and equal to l_p are extracted. Each path is represented by the labels of its nodes. Each path $(v_1, v_2, ..., v_{l_p})$ is then mapped into its corresponding sequence of labels $(l_1, l_2, ..., l_{l_p})$. All the subpaths $\{(v_i, ..., v_j)$ for $1 \leq i \leq j \leq l_p\}$ of a path $(v_1, v_2, ..., v_{l_p})$ are features which will be included in the global index also.

For each extracted path, we keep track also of the number of time it appears in every single graph of the database. All these features are then stored in a Suffix tree. Each node of the tree represents a path obtained during the depth-first search traversal. The path can be reconstructed using its ancestors in the Suffix tree. Each node of the tree also stores the list of graphs containing it together with the number of times the path appears in each graph. The construction and update of the Suffix tree are done during the depth-first search of each graph. GraphGrepSX implements the Suffix tree as an N-ary tree in which the children of a node are represented by a linked list. The list of the occurrences of the features of the graphs is stored in a binary tree indexed by a unique graph id. The Suffix tree and the occurrences list are also stored in an archive file using a compact representation.

The worst case cost to search the child of a given node in the Suffix-Tree is $|l_s|$, where $|l_s|$ is the maximum number of distinct labels in the graph G_i, because the child list is represented as a linked list. Since the list of the feaures occurrences is stored in a binary tree, the cost to update a value is $\log |D|$. The cost of each depth-first visit is $n_i m_i^{l_p}$, where n_i and m_i are respectively the number of the nodes and the maximum valence (degree) of the nodes in the graph G_i. The total cost to build the database index is $O(\sum_i^{|D|}(n_i m_i^{l_p}|l_s|\log|D|))$.

2.2 Filtering and Matching Phases

Given a query graph q, the filtering phase tries to filter out those graphs that cannot match the query graph. This phase is done in two steps. In the first step, the query graph q is processed and its features are extracted and stored in a Suffix Tree. In contrast to the preprocessing phase, here we consider only the maximal paths visited during the depth-first search of the query graph q. A path is considered maximal either if its length is equal to l_p or the path has length less than l_p but cannot be further extended, because the depth-first search can not continue. The nodes of the Suffix tree storing the end-point of a maximal path are marked. Only the occurrences of the maximal paths are stored in the marked nodes of the index.

In the second step the pruning of the candidate graphs of the database is performed by matching the query suffix tree against the suffix tree of the global index. Each marked node of the query tree representing a labeled path $(l_1, l_2, ..., l_n)$ is searched in the Suffix tree of the global index. Those graphs which either do not contain such a path or have such path with an occurrence number less than the occurrence number of the query are discarded. Those that remain represent the candidate set of possibly matching graphs.

The tesing of each candidate graph uses the VF2 [3] library for exhaustive subgraph isomorphism. VF2 is a combinatorial search algorithm which induces a search tree by branching states. It uses a set of topological feasibility rules and a semantic feasibility rule, based on label comparison, to prune the search space. At each state if any rule fails, the algorithm backtracks to the previous step of the match computation.

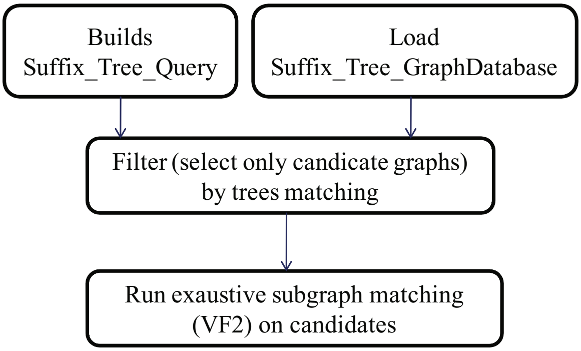

Builds
Suffix_Tree_Query
Load
Suffix_Tree_GraphDatabase
Filter (select only candicate graphs)
by trees matching
Run exaustive subgraph matching
(VF2) on candidates

Fig. 2. Figure shows the filtering phase and the candidates verification phase made by GraphGrepSX

Let T_q to be the query Suffix-Tree and $|T_q|$ the number of nodes inside it.The building cost is $O(n_q m_q^{l_p}|l_s|)$ where n_q is the number of nodes in the tree, m_q is the maximum degree of a node and $|l_s|$ is the maximum number of distinct labels in the query graph. The cost of the pruning step is given by matching time of the query Suffix-Tree against the database index, i.e. $O(|T_q||l_s|)$, plus the average time of the occurrences verification, i.e. $O(|D|\log|D|)$. Therefore the total time is $O(|T_q||l_s||D|\log|D|)$ and, let C be the set of candidates graphs, the cost for each candidate C_i verification is $O(|V[C_i]|!|V[C_i]|)$.

2.3 Experimental Results and Biological Application

GraphGrepSX was implemented in C++ and compiled with the GNU compiler 3.3. In order to evaluate the performance of the proposed approach, it has been compared with the main graph search systems: GraphFind [6], CTree [9], GCoding [16], and SING [5]. Notice that, in what follows, we refer to GraphFind as GraphGrep since we do not use the mining step in the index construction phase. Moreover we do not report comparisons with gIndex [14] since GraphFind outperforms it without using mining. The system has been tested using the Antiviral Screen Dataset [11]. The AIDS database contains the topological structures of 42,000 chemical compounds that have been tested for evidence of anti-HIV activity. It contains sparse graphs having from 20 to 270 nodes. The entire set was divided into three subsets of sizes 8000, 24000 and 42000 respectively. Queries were randomly extracted from the AIDS database selecting a vertex v from a graph of the database and proceeding with a breath-first visit. This process generate groups of 100 queries from each database having a number of edges with 4, 8, 16, and 32 edges. Table 1 shows the index building time for each subset.

Table 1. Index building time (sec)

DB dim.	GraphGrepSX	GraphGrep	CTree	GCoding	SING
8000	16.51	550.4	8.21	632.21	22
24000	38	10399.39	25.34	1956.36	66
42000	66	45600.49	42.42	2944.8	108

GraphGrepSX and CTree yield comparable index construction time and outperform the other approaches. The sizes of the generated indexes are shown in table 2. Thanks to the compactness of its suffix tree structure, GraphGrephSX reduces the redundancy of the index. Therefore GraphGrepSX outperforms the latest graph matching tools. It outperforms SING if used with dynamically changing datasets.

In what follows we show the execution times of the filtering and verification phases. These results report tests made on the entire 42000 AIDS molecular dataset grouped by queries dimension. In table 3 we report the filtering time.

Table 2. Indexes size (Kb)

DB dim.	GraphGrepSX	GraphGrep	CTree	GCoding	SING
8000	3684	293992	13884	6687	8445
24000	11020	928912	41372	20088	22279
42000	18668	1577012	70208	30651	42830

Table 3. Filtering time (sec)

Query dim.	GraphGrepSX	GraphGrep	CTree	GCoding	SING
4	0.05	0.012	1.34	0.0042	0.51
8	0.05	0.006	1.57	0.01	0.76
16	0.041	0.005	1.51	0.026	0.17
32	0.04	0.014	1.01	0.059	0.071

Table 4. Query time (sec)

Query dim.	GraphGrepSX	GraphGrep	CTree	GCoding	SING
4	14.9	15.41	13.27	23.61	12.4
8	17.5	7.1	44.24	15.79	15.24
16	2.08	12.78	51.07	5.39	0.798
32	1.07	3.56	50.91	1.25	0.136

Table 4 shows the total time. The number of generated candidates after the filtering step is shown in table 5. CTree and GCoding generate smaller candidates sets. This is due to the fact that such indexes are able to capture the structure of the graphs. Unfortunately, they require more execution time because of the approximate match on the closure graphs and the mining operations during the filtering step.

Table 5. Number of generated candidates

Query dim.	GraphGrepSX	GraphGrep	CTree	GCoding	SING
4	26865	29196	16704	16188	23170
8	21337	13920	5840	8567	14012
16	1629	7053	289	1648	214
32	142	3193	3	142	4

Table 6. Total time time (sec)

Query dim.	GraphGrepSX	SING
4	15.89	23.39
8	18.56	26.42
16	3.07	1f1.45
32	2.04	10.68

GraphGrepSX and GraphGrep uses the same matching algorithm, but the first generates a smaller number of candidates by applying a redundant check deletion phase.

In Table 6 we report the total time needed by GraphGrepSX and SING to execute a single query. SING has an overhead of 10.5 seconds to load the index. Whereas GraphGrepSX needs less than one second (0.93 seconds) to load the index.

3 Conclusion

Indexing paths instead of subgraphs may result in more preprocessing time and indexing space. However, paths require less filtering and querying time. Results show that a further improvement on path-index base system is achieved by making use of Suffix Trees. GraphGrephSX reduces the size and time needed for the construction of large database index compared to the most prominent graph querying systems. Furthermore, GraphGrephSX outperforms all compared systems when the index structure needs to be rebuilt. It can be considered to be a good compromise between preprocessing time and querying time.

References

1. Cheng, J., Ke, Y., Ng, W., Lu, A.: Fg-index: towards verification-free query processing on graph databases. In: Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 857–872 (2007)
2. Cohen, E., Datar, M., Fujiwara, S., Gionis, A., Indyk, P., Motwani, R., Ullman, J.D., Yang, C.: Finding interesting associations without support pruning. IEEE Transactions on Knowledge and Data Engineering 13(1), 64–78 (2001)
3. Cordella, L., Foggia, P., Sansone, C., Vento, M.: A (sub)graph isomorphism algorithm for matching large graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(10), 1367–1372 (2004)
4. Daylight chemical information systems, `http://www.daylight.com/`
5. Di Natale, R., Ferro, A., Giugno, R., Mongiovi, M., Pulvirenti, A., Shasha, D.: Sing: Subgraph search in non-homogeneous graphs. BMC bioinformatics 11(1), 96 (2010)
6. Ferro, A., Giugno, R., Mongiovì, M., Pulvirenti, A., Skripin, D., Shasha, D.: Graphfind: enhancing graph searching by low support data mining techniques. BMC bioinformatics 9(suppl. 4), S10 (2008)

7. Frowns, http://frowns.sourceforge.net/
8. Giugno, R., Shasha, D.: Graphgrep: A fast and universal method for querying graphs. In: Proceeding of the International Conference in Pattern Recognition (ICPR), pp. 112–115 (2002)
9. He, H., Singh, A.K.: Closure-tree: An index structure for graph queries. In: Proceedings of the 22nd International Conference on Data Engineering, ICDE 2006, p. 38 (2006)
10. Messmer, B.T., Bunke, H.: Subgraph isomorphism detection in polynominal time on preprocessed model graphs. In: Proceedings of Asian Conference on Computer Vision, pp. 373–382 (1995)
11. National Cancer Institute. U.S. National Institute of Health, http://www.cancer.gov/
12. Shasha, D., Wang, J.T.-L., Giugno, R.: Algorithmics and applications of tree and graph searching. In: Proceeding of the ACM Symposium on Principles of Database Systems (PODS), pp. 39–52 (2002)
13. Ukkonen, E.: Approximate string-matching over suffix trees. In: Combinatorial Pattern Matching, pp. 228–242. Springer, Heidelberg (1993)
14. Yan, X., Yu, P.S., Han, J.: Graph indexing based on discriminative frequent structure analysis. ACM Transactions on Database Systems 30(4), 960–993 (2005)
15. Zhang, S., Hu, M., Yang, J.: Treepi: A novel graph indexing method. In: Proceedings of IEEE International Conference on Data Engineering, pp. 966–975 (2007)
16. Zou, L., Chen, L., Yu, J.X., Lu, Y.: A novel spectral coding in a large graph database. In: Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology, pp. 181–192. ACM, New York (2008)

Part III

Learning Methods for Gene Expression and Mass Spectrometry Data

Semi-Supervised Graph Embedding Scheme with Active Learning (SSGEAL): Classifying High Dimensional Biomedical Data

George Lee and Anant Madabhushi

Rutgers, The State University of New Jersey,
Department of Biomedical Engineering
Piscataway, NJ 08854 USA
geolee@eden.rutgers.edu, anantm@rci.rutgers.edu

Abstract. In this paper, we present a new dimensionality reduction (DR) method (SSGEAL) which integrates Graph Embedding (GE) with semi-supervised and active learning to provide a low dimensional data representation that allows for better class separation. Unsupervised DR methods such as Principal Component Analysis and GE have previously been applied to the classification of high dimensional biomedical datasets (e.g. DNA microarrays and digitized histopathology) in the reduced dimensional space. However, these methods do not incorporate class label information, often leading to embeddings with significant overlap between the data classes. Semi-supervised dimensionality reduction (SSDR) methods have recently been proposed which utilize both labeled and unlabeled instances for learning the optimal low dimensional embedding. However, in several problems involving biomedical data, obtaining class labels may be difficult and/or expensive. SSGEAL utilizes labels from instances, identified as “hard to classify” by a support vector machine based active learning algorithm, to drive an updated SSDR scheme while reducing labeling cost. Real world biomedical data from 7 gene expression studies and 3900 digitized images of prostate cancer needle biopsies were used to show the superior performance of SSGEAL compared to both GE and SSAGE (a recently popular SSDR method) in terms of both the Silhouette Index (SI) (SI = 0.35 for GE, SI = 0.31 for SSAGE, and SI = 0.50 for SSGEAL) and the Area Under the Receiver Operating Characteristic Curve (AUC) for a Random Forest classifier (AUC = 0.85 for GE, AUC = 0.93 for SSAGE, AUC = 0.94 for SSGEAL).

1 Introduction

Dimensionality reduction (DR) is useful for extracting a few relatively simple patterns from more complex data. For very high dimensional data, such as gene expression, the original feature space could potentially span up to tens of thousands of features. This makes it difficult to build generalizable predictors on account of the curse of dimensionality problem [1], where the feature space is much larger than the number of samples available for classifier training. Therefore, DR methods are often utilized as a precursor to classification. Predictors

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 207–218, 2010.

can then be trained on low dimensional embedded features, resulting in improved classification accuracy while also allowing researchers to visualize and interpret relationships between data points [1].

Most commonly used DR methods, such as Principal Component Analysis (PCA) [2], Graph Embedding [3], or Manifold Learning [2] schemes are unsupervised, meaning they do not take into account class label information. These methods essentially use cost functions assuming that the best features lie in a subspace of the original high dimensional space where most of the variance in the data is centered. Supervised DR methods such as linear discriminant analysis (LDA) [1] employ cost functions where class labels are incorporated to help separate known classes in a low dimensional embedding.

LDA is one of the most popular supervised DR methods; however it does not consider unlabeled instances [1, 4]. Blum et al. [5] suggested that incorporating unlabeled samples in addition to labeled samples can significantly improve classification results. Subsequently, many new DR methods employ semi-supervised (SS) or weakly labeled learning techniques which incorporate the use of both labeled and unlabeled data [4, 6–9]. These SSDR schemes use labeled information in the construction of a pairwise similarity matrix, where the individual cells are assigned weights based on class and feature-based similarity between sample pairs. These weights can then be used to create a low dimensional mapping by solving a simple eigen-problem, the hypothesis being that embeddings explicitly employing label information result in greater class separation in the reduced dimensional space.

Active Learning (AL) algorithms have been utilized to intelligently identify hard to classify instances. By querying labels for only hard to classify instances, and using them to train a classifier, the resulting classifier has higher classification accuracy compared to random learning, assuming the same number of queries are used for classifier training [10, 11]. In practice, obtaining labels for biomedical data is often expensive. For example, in the case of digital pathology applications, disease extent can only be reliably annotated by an expert pathologist. By employing AL, the predictive model is (a) cheaper to train and (b) yields a superior decision boundary for improved discrimination between object classes with fewer labeled instances.

In this paper we present Semi-Supervised Graph Embedding with Active Learning (SSGEAL), a new DR scheme for analysis and classification of high dimensional, weakly labeled data. SSGEAL identifies the most difficult to classify samples via a support vector machine based active learning scheme, which is then used to drive a semi-supervised graph embedding algorithm. Predictors can then be trained for object classification in the SSGEAL reduced embedding space.

2 Previous Work and Novel Contributions

2.1 Unsupervised Dimensionality Reduction

PCA is the most commonly used unsupervised DR method. However it is essentially a linear DR scheme [2]. Nonlinear dimensionality reduction (NLDR) methods such as Isomap [2] and Locally Linear Embedding [2], are powerful due to their ability to discover nonlinear relationships between samples. In [1], we found that nonlinear DR schemes outperformed PCA for the problem of classifying high dimensional gene- and protein-expression datasets. However, NLDR schemes are notoriously unstable [1, 2], requiring careful tuning of a neighborhood parameter to generate useful embeddings.

Graph Embedding [3], or Spectral Embedding is an alternative unsupervised NLDR method which does not require adjusting a neighborhood parameter, and has been found to be useful in applications involving classification of DNA microarrays, proteomic spectra, and biomedical imaging [1, 12]. Normalized cuts [3] is one implementation of Graph Embedding, which is widely used in the area of image segmentation. Other versions of graph embedding include Min Cut [5], Average Cut [3], Associative Cut [3], and Constrained Graph Embedding [13].

2.2 Semi-Supervised Dimensionality Reduction

Sugiyama et al. [4] applied SS-learning to Fisher's discriminant analysis in order to find projections that maximize class separation. Yang et al. [8] similarly applied SS-learning toward manifold learning methods. Sun et al. [9] implemented a SS version of PCA by exploiting between-class and within-class scatter matrices. SSAGE [6] is a SS method for spectral clustering which utilizes weights to simultaneously attract within-class samples and repel between-class samples given a neighborhood constraint. However, these embeddings often contain unnatural, contrived clusters on account of labeled samples. Zhang [7] uses a similar approach to SSDR, but without utilizing neighborhood constraints.

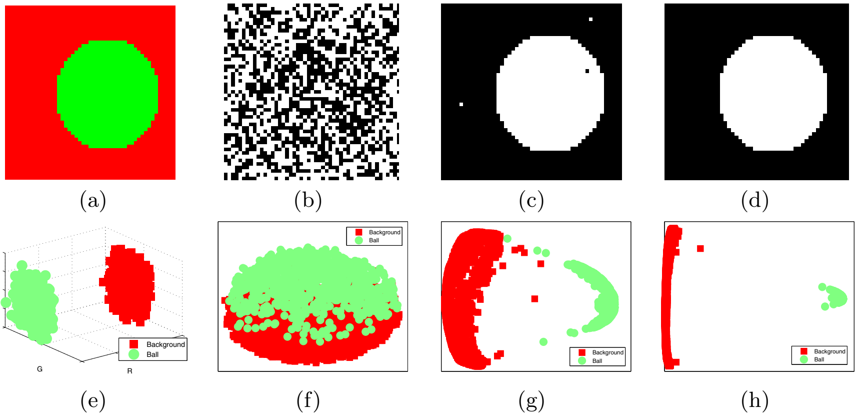
2.3 Active Learning

Previous AL methods have looked at the variance of sample classes to identify difficult to classify instances [14]. The Query by Committee approach [10] uses disagreement across several weak classifiers to identify hard to classify samples. In [15], a geometrically based AL approach utilized support vector machines (SVMs) to identify confounding samples as those that lay closest to the decision hyperplane. SVM-based AL has previously been applied successfully to the problem of classifying gene expression data [11]. Additionally, a clear and easily interpretable rationale for choice of sample selection exists. All these methods however have typically been applied to improving classification and not embedding quality per se [10, 14].

2.4 Novel Contributions and Significance of SSGEAL

The primary contribution of this paper is that it merges two powerful schemes - SSDR with Active Learning - for generating improved low dimensional embedding representations, which allows for greater class separation.

Figure 1 illustrates how Graph Embedding (GE) can be improved with SS-learning (SSAGE), and even further using AL (SSGEAL). In Figure 1(a), a simple RGB image consisting of ball and background pixels is shown. Following the addition of Gaussian noise, each pixel in Figure 1a is plotted in a 3D RGB space (Figure 1(e)). Subsequently, we reduce the 3D RGB space into a 2D embedding via GE (Figure 1(f)), SSAGE (Figure 1(g)), and SSGEAL (Figure 1(h)). Figures 1(b), 1(c), and 1(d) represent a pixel-wise binary classification into foreground (ball) and background classes via GE, SSAGE, and SSGEAL, respectively. These were obtained via replicated k-means clustering on the corresponding DR embeddings, as shown in Figures 1(f), 1(g), and 1(h).



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 1. (a) RGB image containing ball against colored background pixels. (e) image pixels plotted in 3D RGB space. The binary classifications (b-d) reflect the corresponding quality of embeddings obtained via DR methods (b) GE, (c) SSAGE, and (d) SSGEAL. These were obtained via replicated k-means clustering on the reduced embeddings by (f) GE, (g) SSAGE, and (h) SSGEAL, respectively.

Table 1. Commonly used notation in this paper

Symbol	*Description*
X	Set containing N samples
$\mathbf{x}_i$, $\mathbf{x}_j$	Sample vector $\mathbf{x}_i$, $\mathbf{x}_j \in X$, $i, j \in \{1, 2, ..., N\}$, $\mathbf{x} \in \mathbb{R}^n$
n	Number of features used to describe $\mathbf{x}_i$
W	Dissimilarity matrix
$Y(\mathbf{x}_i)$	Labels for samples $\mathbf{x}_i$, $Y(\mathbf{x}_i) \in \{+1, -1\}$
$Z(X, Y(X_{Tr}))$	Embedding Z constructed using data X and label set $Y(X_{Tr})$.
X_{Tr}	Set of labeled training samples $\mathbf{x}_i \in X_{Tr}$
X_{Ts}	Set of unlabeled testing samples $X_{Ts} \subset X$
X_a	Set of ambiguous samples $X_a \subset X_{Ts}$
δ	Distance to decision hyperplane F in SVM-based AL

3 Review of SSDR and Active Learning Methods

3.1 Graph Embedding (GE)

To obtain low dimensional embedding Z, Graph Embedding [3] utilizes pairwise similarities between objects $\mathbf{x}_i$ and $\mathbf{x}_j \in X$ to construct $N \times N$ weighted graph

$$W(\mathbf{x}_i, \mathbf{x}_j) = e^{\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\sigma}}, \tag{1}$$

where $\sigma = 1$. $\tilde{W}(\mathbf{x}_i, \mathbf{x}_j) = (\sum_{ii}^{N} W(\mathbf{x}_{ii}, \mathbf{x}_j) \times \sum_{jj}^{N} W(\mathbf{x}_i, \mathbf{x}_{jj}))^{-1} W(\mathbf{x}_i, \mathbf{x}_j)$ is then used to solve the eigenvalue problem $(D - \tilde{W})\mathbf{z} = \lambda D \mathbf{z}$, where D is a diagonal matrix containing the trace of $\tilde{W}$, and $\mathbf{z}_k$ are the eigenvectors. Embedding Z is formed by taking the most dominant eigenvectors $\mathbf{z}_k$ corresponding to the k smallest eigenvalues λ_k, where k is the dimensionality of Z. In this implementation, Graph Embedding does not consider labeled information.

3.2 Semi-Supervised Agglomerative Graph Embedding (SSAGE)

By using known label information, Zhao [6], describes a method for SSDR where the similarity weights for GE are adjusted such that Equation 1 is replaced by

$$W(\mathbf{x}_i, \mathbf{x}_j) \quad = \quad \begin{cases} (e^{\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\sigma}})(1 + e^{\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\sigma}}), & \text{if } Y(\mathbf{x}_i) = Y(\mathbf{x}_j) \\ (e^{\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\sigma}})(1 - e^{\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\sigma}}), & \text{if } Y(\mathbf{x}_i) \neq Y(\mathbf{x}_j) \\ e^{\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\sigma}} \quad , & \text{otherwise} \end{cases} \tag{2}$$

In contrast to simple GE, in SSAGE, known labeled samples are mapped to be closer in the embedding space Z if both samples $\mathbf{x}_i$ and $\mathbf{x}_j$ are of the same class $Y(\mathbf{x}_i) = Y(\mathbf{x}_j)$, and further apart if both samples are of different classes.

3.3 SVM-Based Active Learning to Identifying Ambiguous Samples

A labeled set X_{Tr} is first used to train the SVM. SVMs [16] project the input training data onto a high-dimensional space using the kernel $\Pi(\mathbf{x}_i, \mathbf{x}_j)$. A linear

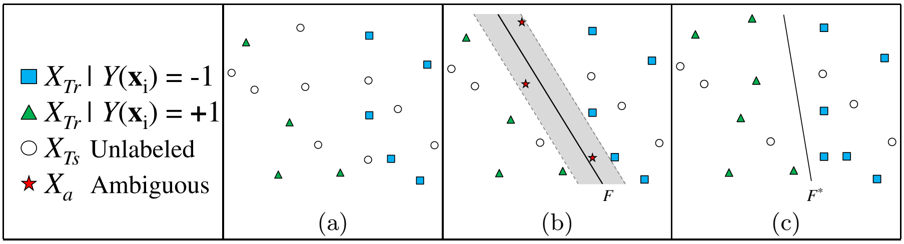

■ X_{Tr} | $Y(\mathbf{x}_i)$ = -1
▲ X_{Tr} | $Y(\mathbf{x}_i)$ = +1
○ X_{Ts} Unlabeled
★ X_a Ambiguous
F
F*
(a)
(b)
(c)

Fig. 2. (a) Labeled samples $\mathbf{x}_i \in X_{Tr}$ are used to train an SVM model F. (b) Unlabeled samples $\mathbf{x}_i \in X_{Ts}$ found to be mapped closest to the model hyperplane F are included into set X_a. (c) Labels Y($\mathbf{x}_i \in X_a$) are queried and used to improve the new SVM model F^*, yielding a better predictor compared to F.

kernel, defined as $\Pi(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$, can then be used to maximize the margins so as to decrease prediction risk. A decision boundary F is created in the trained feature space by maximizing the margin between classes. Unlabeled instances $\mathbf{x}_i \in X_{Ts}$ are mapped into the same feature space (Figure 2(a)).

However, instead of classifying X_{Ts}, we use boundary F to find ambiguous samples $\mathbf{x}_i \in X_a$ via measure δ, defined as the relative distance to hyperplane F. Samples $\mathbf{x}_i \in X_{Ts}$ of shortest δ represent the most ambiguous samples and are assigned to set X_a (Figure 2(b)). Labels for X_a are queried and these ambiguous samples are added to the subsequent training set $X_{Tr} = [X_{Tr}, X_a]$. Learning via the updated labels $Y(X_{Tr})$ results in improved class separation (Figure 2(c)).

4 Semi-Supervised Graph Embedding with Active Learning (SSGEAL)

4.1 Initialization with Initial Embedding Z_0

The schema for SSGEAL is illustrated via the flowchart in Figure 3. Our initialization comprises of creating an initial embedding Z_0 and defining the initial training X_{Tr} for our active learning scheme within Z_0. Given data set X, we use Graph Embedding as illustrated in Section 3.1 to obtain our initial embedding $Z_0(X) = [\mathbf{z}_1, ..., \mathbf{z}_k]$, or simply Z_0.

4.2 Active Learning to Identify Ambiguous Samples $\mathbf{X}_a$

SVM-based active learning (see Section 3.3) is used to identify ambiguous samples $\mathbf{x}_i \in X_a$ in embedding Z_q, where q represents the specific iteration of an embedding Z. Initial labeled training samples X_{Tr} for AL are selected randomly from X. We begin by training an SVM using $Z_0(X_{Tr})$ and $Y(X_{Tr})$ to create model F. $\delta(X_{Ts})$ can be found using F, where the smallest $\delta(X_{Ts})$ are selected and assigned to set X_a. $Y(X_a)$ is revealed and X_a is added to the training set X_{Tr}, such that $X_{Tr} = [X_{Tr}, X_a]$.

4.3 Semi-Supervised Graph Embedding Z_q Using Updated Labels

We utilize an updated version of Zhao's SSAGE method [6] to map a modified similarity matrix W into Z using the GE framework discussed in Section 3.1. This weighting only takes into account samples which are of the same class, using a gravitation constant $G > 1$ to attract same-class samples closer. Weights are adjusted such that Equation 1 is replaced by

$$W(\mathbf{x}_i, \mathbf{x}_j) \quad = \quad \begin{cases} G \times e^{\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\sigma}}, & \text{if } Y(\mathbf{x}_i) = Y(\mathbf{x}_j) \\ e^{\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2}{\sigma}}, & \text{otherwise} \end{cases} \tag{3}$$

Unlike the Zhao [6] and Zhang [7] implementations, instances from different classes are not explicitly weighted to force them farther apart in SSGEAL. The rationale for this is that for biomedical data, certain instances within one class may share several traits with another class. For instance, premalignant lesions while technically benign, share several hallmarks of malignant tumors. Artificially forcing instances from different classes farther apart could result in a pre-malignant lesion being mapped far apart from the cancer class, rather than in an intermediate class between benign and malignant.

Labels $Y(X_{Tr})$ from the updated training set and current embedding Z_q are used to create embedding Z_{q+1}. The new embedding $Z_{q+1}(Z_q, Y(X_{Tr}))$, or simply Z_{q+1}, is constructed using the current embedding Z_q and the exposed label set $Y(X_{Tr})$. The process of obtaining new labels from AL and creating semi-supervised embeddings continues until the stopping criterion is met.

4.4 Stopping Criterion Using Silhouette Index

The stopping criterion is set using the Silhouette Index (ϕ^{SI}) [17] of the revealed labels. ϕ^{SI} is a cluster validity measure which captures the intra-cluster compactness $A_i = \sum_{j,Y(\mathbf{x}_j)=Y(\mathbf{x}_i)} \|\mathbf{x}_i - \mathbf{x}_j\|_2$, which represents the average distance of a point $\mathbf{x}_i$ from other points X_j of the same class, while also taking into account inter-cluster separation $B_i = \sum_{j,Y(\mathbf{x}_j)\neq Y(\mathbf{x}_i)} \|\mathbf{x}_i - \mathbf{x}_j\|_2$, the minimum of the average distances of a point $\mathbf{x}_i$ from other instances in different classes. Thus, the formulation for Silhouette Index is shown as

$$\phi^{SI} = \sum_{i}^{N} \frac{B_i - A_i}{\max[A_i, B_i]}. \tag{4}$$

ϕ^{SI} ranges from -1 to 1, where -1 is the worst, and 1 is the best possible clustering. When the change in ϕ^{SI} falls below threshold θ, such that $|\phi^{SI}_{q+1} - \phi^{SI}_q| < \theta$, the algorithm stops. The algorithm for SSGEAL is presented below.

```
Algorithm SSGEAL
Input: $X$,$Y(X_{Tr})$ $\theta$, $\delta$, $q = 0$
Output: $Z_f$
begin
  0. Build initial embedding $Z_0(X)$
  1. while $|\phi^{SI}_{q+1} - \phi^{SI}_q| < \theta$
  2.     Train SVM model $F$ using $X_{Tr}, Y(X_{Tr})$
  3.     Identify $\mathbf{x}_i \in X_a$ using measure $\delta$
  4.     Update $X_{Tr} = [X_{Tr}, X_a]$
  5.     Update embedding $Z_{q+1}(Z_q, Y(X_{Tr}))$ via Equation 3
  6.     Compute $\phi^{SI}_q$ using Equation 4.
  7.     $q = q + 1$
  8. endwhile
  9. return $Z_f$
end
```

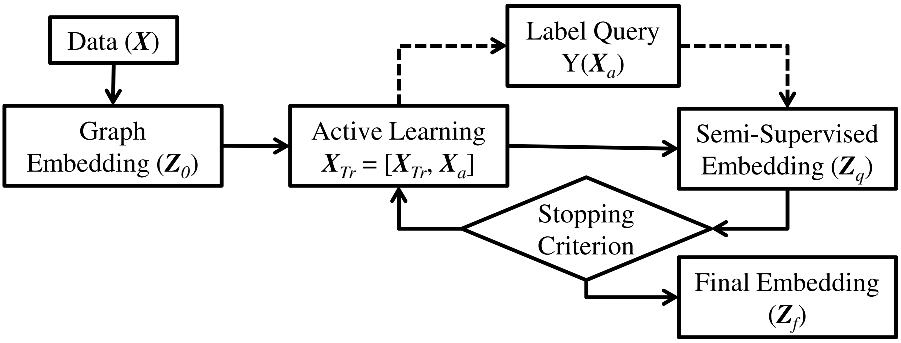

Data ($\boldsymbol{X}$)
Graph Embedding ($\boldsymbol{Z}_0$)
Active Learning $\boldsymbol{X}_{Tr} = [\boldsymbol{X}_{Tr}, \boldsymbol{X}_a]$
Label Query $Y(\boldsymbol{X}_a)$
Semi-Supervised Embedding ($\boldsymbol{Z}_q$)
Stopping Criterion
Final Embedding ($\boldsymbol{Z}_f$)

Fig. 3. Flowchart of SSGEAL

5 Experimental Results and Discussion

5.1 Experiments and Evaluation

Datasets. Table 2 provides an overview of the 7 publically available gene expression and digitized prostate biopsy images used to test SSGEAL.[1] For the gene expression datasets, no preprocessing or normalization of any kind was performed prior to DR. For the Prostate Histopathology dataset, a set of 14 pixel-wise features were extracted, including first-order statistical, second-order co-occurrence, and steerable Gabor wavelet features [10, 18] from the images, digitized at 40x magnification. The images are then broken into 30 x 30 pixel regions, each quantified by averaging the feature values in the region. We randomly selected 3900 non-overlapping patches from within the cancer and non-cancer regions (manually annotated by an expert pathologist) for purposes of evaluation.

Table 2. Datasets used in our experiments

	Datasets	Description
Gene Expression	Prostate Cancer	25 Tumor, 9 Normal, 12600 genes
	Colon Cancer	22 Tumor, 40 Normal, 2000 genes
	Lung Cancer	15 MPM, 134 ADCA, 12533 genes
	ALL / AML	20 ALL, 14 AML, 7129 genes
	DLBCL Tumor	58 Tumor, 19 Normal, 6817 genes
	Lung Cancer(Mich)	86 Tumor, 10 Normal, 7129 genes
	Breast Cancer	10 Tumor, 20 Normal, 54675 genes
Imaging	Prostate Histopathology	1950 cancer regions, 1950 benign regions, 14 image textural descriptors

[1] Gene expression datasets were obtained from the Biomedical Kent-Ridge Repositories at http://sdmc.lit.org.sg/GEDatasets/Datasets and http://sdmc.i2r.a-star.edu.sg/rp

Table 3. $\mu(\phi^{SI})$, $\mu(\phi^{AUC})$, $\sigma(\phi^{SI})$, and $\sigma(\phi^{AUC})$ across 10 runs using different X_{Tr} for GE, SSAGE and SSGEAL. The high mean performance and low standard deviation of these statistics over 10 runs of SSGEAL on 8 datasets demonstrates the robustness of the algorithm regardless of initial training set X_{Tr}. Best values are shown in **bold**. For a majority of the cases, SSGEAL is shown to perform the best.

		Silhouette Index			Random Forest AUC		
	Datasets	GE	SSAGE	SSGEAL	GE	SSAGE	SSGEAL
Gene Expression	Prostate Cancer	0.54	0.29±0.10	**0.66**±0.01	**1.00**	0.98±0.04	**1.00**±0.00
	Colon Cancer	0.02	0.16±0.01	**0.43**±0.04	0.73	0.92±0.03	**0.95**±0.05
	Lung Cancer	0.64	0.49±0.06	**0.65**±0.20	0.49	0.95±0.10	**0.96**±0.09
	ALL / AML	0.42	0.24±0.04	**0.47**±0.05	0.95	0.96±0.03	**0.97**±0.04
	DLBCL Tumor	0.20	0.32±0.10	**0.62**±0.03	0.75	0.89±0.04	**0.95**±0.04
	Lung Cancer(Mich)	0.68	0.45±0.02	**0.83**±0.02	**1.00**	0.95±0.13	0.99±0.03
	Breast Cancer	0.20	0.19±0.09	**0.45**±0.08	0.78	0.90±0.05	**0.96**±0.05
Imaging	Prostate Histopathology	0.35	**0.36**±0.00	0.35±0.00	0.85	**0.93**±0.00	**0.93**±0.00

Experiments. Two DR techniques were employed to compete against our algorithm (SSGEAL): one which does not incorporate labels (GE) and one which utilizes labels (SSAGE). We generated embeddings Z using DR methods GE, SSAGE, and SSGEAL to show that (a) embeddings generated using SSGEAL outperform those generated via GE and SSAGE, (b) steady improvement in both classification accuracy and Silhouette index can be observed via active learning with SSGEAL, and (c) SSGEAL is robust to initial training.

Evaluation Measures. Embeddings were evaluated both qualitatively and quantitatively using ϕ^{SI} (Equation 4) and Area Under the Receiver Operating Characteristic (ROC) Curve for Random Forest Classification ϕ^{AUC}. For ϕ^{SI}, all labels were used. For ϕ^{AUC}, a randomly selected training pool $\mathcal{P}$ consisting of two-thirds of the instances in X was used, with the remaining samples reserved for testing. 50 decision trees were trained using a 50 random subsets each consisting of 2/3 of $\mathcal{P}$. Predictions on the testing samples were subsequently bagged and used to calculate the ROC curve for assessing classifier performance.

Parameter Settings. For our experiments, 2D embeddings $Z = [\mathbf{z}_1, \mathbf{z}_2]$ are generated for each DR method. In all cases, no neighborhood information was used. For both SSAGE and SSGEAL, we ultimately expose 40% of the labels. For SSGEAL, the gravitation constant G was set to 1.3 and our initial training set X_{Tr} was set at 15% of $Y(X)$, revealing 5% of the labels $Y(X_{Ts})$ at each iteration q until 40% of the labels were revealed.

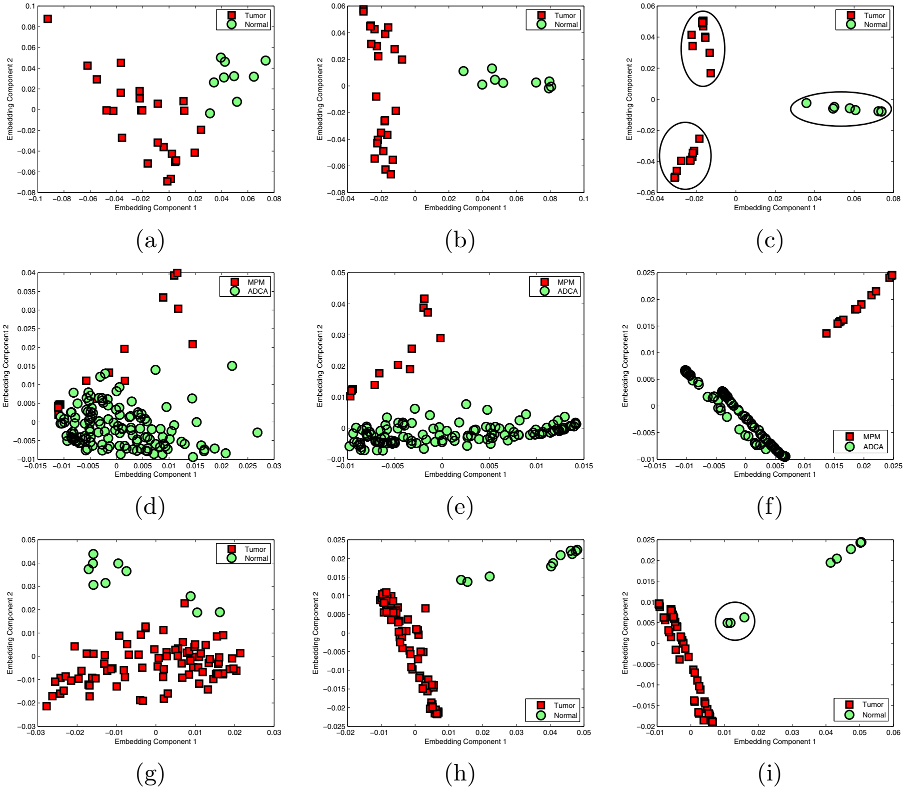
5.2 Comparing SSGEAL with GE and SSAGE via ϕ^{SI} and ϕ^{AUC}

Table 3 lists the mean and variance of ϕ^{AUC} and ϕ^{SI} values for SSGEAL, GE, and SSAGE, over 8 dataset. The same number of labeled samples (40%) were used for SSAGE and SSGEAL for each data set. To obtain an accurate representation of algorithm performance, we randomly selected 10 training sets X_{Tr} for

10 runs of SSAGE and SSGEAL for the purpose of testing the robustness of the algorithms to initial labeling. Note that GE is an unsupervised method and does not utilize label information, hence there is no standard deviation across multiple runs of GE. 2D embeddings were generated for each set X_{Tr} and evaluated via ϕ^{AUC} and ϕ^{SI}.

For a majority of the datasets, SSGEAL outperforms both GE and SSAGE in terms of ϕ^{SI} ($\mu(\phi^{SI})$ of 0.35 for GE, 0.31 for SSAGE, and 0.50 for SSGEAL) and ϕ^{AUC} ($\mu(\phi^{AUC})$ of 0.85 for GE, 0.93 for SSAGE, and 0.94 for SSGEAL). Furthermore, low standard deviation ($\sigma(\phi^{AUC})$, $\sigma(\phi^{SI})$) over the 10 runs suggest robustness of SSGEAL to initial X_{Tr}.

Figure 4 shows qualitative illustrations of 2D embeddings for GE and SSGEAL over different iterations for 3 selected datasets. We can observe greater class separation and cluster tightness with increasing iterations for SSGEAL.



(a) (b) (c)

(d) (e) (f)

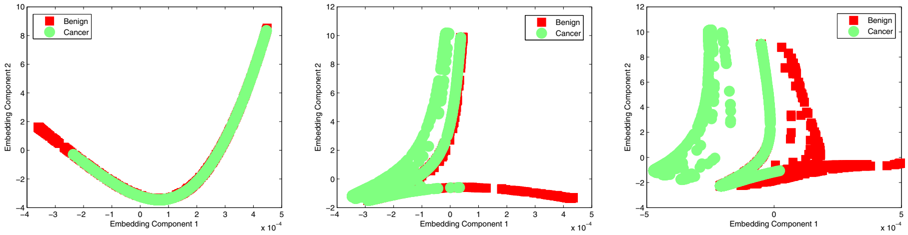
(g) (h) (i)

Fig. 4. Scatter plots of the 2 most dominant embedding eigenvectors $\mathbf{z}_1(\mathbf{x}_i)$, $\mathbf{z}_2(\mathbf{x}_i)$ for $\mathbf{x}_i \in X$ are shown for different iterations of SSGEAL (a) Z_0, (b) Z_2, and (c) Z_f (the final stable embedding), for the Prostate Cancer dataset. Similarly, the embedding plots are shown for the Lung Cancer dataset for (d) Z_0, (e) Z_2, (f) Z_f. Lastly, (g) Z_0, (h) Z_2, (i) Z_f are shown for the Lung Cancer(Mich) dataset. Note the manually placed ellipses in (c) and (i) highlight what appear to be novel subclasses.

Figures 4(a), 4(d), and 4(g) show embedding plots of GE (Z_0). An intermediate step of SSGEAL (Z_q) is shown in Figures 4(b), 4(e), and 4(h) and SSGEAL embeddings (Z_f) can be seen in Figures 4(c), 4(f), and 4(i).

6 Concluding Remarks

Semi-Supervised Graph Embedding with Active Learning (SSGEAL) represents the first attempt at incorporating an active learning algorithm into a semi-supervised dimensionality reduction (SSDR) framework. The inclusion of active learning is especially important for problems in biomedical data where class labels are often difficult or expensive to come by. Using 8 real-world gene expression and digital pathology image datasets, we have shown that SSGEAL results in low dimensional embeddings which yield tighter, more separated class clusters and result in greater class discriminability compared to GE and SSAGE, as evaluated via the Silhouette Index and AUC measures. Furthermore, SSGEAL was found to be robust with respect to the choice of initial labeled samples used for initializing the active learning process. SSGEAL does however appear to be sensitive to the value assigned to the gravitation constant G. This parameter may be used to refine the initial graph embedding (Figure 5(a)). For the histology dataset, setting $G = 1.5$ resulted in $\phi^{SI} = 0.39$ and $\phi^{AUC} = 0.94$ for SSGEAL, compared to $\phi^{SI} = 0.36$ and $\phi^{AUC} = 0.93$ for SSAGE (Figure 5). In future work we intend to extensively and quantitatively evaluate the sensitivity of our scheme to neighborhood, gravitation, and stopping parameters.



(a) (b) (c)

Fig. 5. Scatter plots of the 2 most dominant embedding eigenvectors are shown for the Prostate Histopathology dataset. (b) and (c) show SSGEAL embeddings with gravitation constants $G = 1.3$ and 1.5 respectively, suggesting the utility of G for improving embeddings with large degrees of overlap between the object classes. For comparison, the embedding graph for GE is also shown for this dataset (Figure 5(a)).

Acknowledgments. Funding for this work made possible by the Wallace H. Coulter Foundation, New Jersey Commission on Cancer Research, National Cancer Institute (R01CA136535-01, R21CA127186-01, R03CA128081-01), the Cancer Institute of New Jersey, and Bioimagene Inc.

References

1. Lee, G., Rodriguez, C., Madabhushi, A.: Investigating the efficacy of nonlinear dimensionality reduction schemes in classifying gene and protein expression studies. IEEE Trans. on Comp. Biol. and Bioinf. 5(3), 368–384 (2008)
2. van der Maaten, L.J.P., Postma, E.O., van den Herik, H.J.: Dimensionality reduction: A comparative review. Tilburg University Technical Report, TiCC- TR2009–005 (2009)
3. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Analysis and Machine Intelligence. 22(8), 888–905 (2000)
4. Sugiyama, M., Idé, T., Nakajima, S., Sese, J.: Semi-supervised local fisher discriminant analysis for dimensionality reduction. Advances in Knowledge Discovery and Data Mining, 333–344 (2008)
5. Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph mincuts. In: International Conference on Machine Learning, pp. 19–26 (2001)
6. Zhao, H.: Combining labeled and unlabeled data with graph embedding. Neurocomputing 69(16-18), 2385–2389 (2006)
7. Zhang, D., et al.: Semi-supervised dimensionality reduction. In: SIAM International Conference on Data Mining (2007)
8. Yang, X., Fu, H., Zha, H., Barlow, J.: Semi-supervised nonlinear dimensionality reduction. In: International Conference on Machine Learning, pp. 1065–1072 (2006)
9. Sun, D., Zhang, D.: A new discriminant principal component analysis method with partial supervision. Neural Processing Letters 30, 103–112 (2009)
10. Doyle, S., et al.: A class balanced active learning scheme that accounts for minority class problems: Applications to histopathology. In: MICCAI (2009)
11. Liu, Y.: Active learning with support vector machine applied to gene expression data for cancer classification. J. Chem. Inf. Comput. Sci. 44(6), 1936–1941 (2004)
12. Higgs, B.W., et al.: Spectral embedding finds meaningful (relevant) structure in image and microarray data. BMC Bioinformatics 7(74) (2006)
13. He, X., Ji, M., Bao, H.: Graph embedding with constraints. In: International Joint Conference on Artificial Intelligence, pp. 1065–1070 (2009)
14. Cohn, D.A., Ghahramani, Z., Jordan, M.I.: Active learning with statistical models. Journal of Artif. Intell. Res. 4, 129–145 (1996)
15. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. Journal of Machine Learning Research, 999–1006 (2000)
16. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learning 20 (1995)
17. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. 20(1), 53–65 (1987)
18. Doyle, S., Tomaszewski, J., Feldman, M., Madabhushi, A.: Hierarchical boosted bayesian ensemble for prostate cancer detection from digitized histopathology. IEEE Transactions on Biomedical Engineering (2010)

Iterated Local Search for Biclustering of Microarray Data

Wassim Ayadi[1,2], Mourad Elloumi[2], and Jin-Kao Hao[1]

[1] LERIA, University of Angers, 2 Boulevard Lavoisier, 49045 Angers, France
[2] UTIC, Higher School of Sciences and Technologies of Tunis, 1008 Tunis, Tunisia
{ayadi,hao}@info.univ-angers.fr, mourad.elloumi@fsegt.rnu.tn

Abstract. In the context of microarray data analysis, biclustering aims to identify simultaneously a group of genes that are highly correlated across a group of experimental conditions. This paper presents a Biclustering Iterative Local Search (BILS) algorithm to the problem of biclustering of microarray data. The proposed algorithm is highlighted by the use of some original features including a new evaluation function, a dedicated neighborhood relation and a tailored perturbation strategy. The BILS algorithm is assessed on the well-known yeast cell-cycle dataset and compared with two most popular algorithms.

Keywords: Analysis of DNA microarray data, biclustering, evaluation function, iterative local search.

1 Introduction

With the fast advances of DNA Microarray technologies, more and more gene expression data are made available for analysis. In this context, *biclustering* has been recognized as a remarkably effective method for discovering several groups of subset of genes associated with a subset of conditions. These groups are called *biclusters*. Biclusters can be used for various purposes, for instance, they are useful to discover genetic knowledge, such as gene annotation or gene interaction, and to understand various genetic diseases.

Formally, DNA microarray data is usually represented by a data matrix $M(I, J)$, where the i^{th} row, $i \in I=\{1, 2, \dots, n\}$, represents the i^{th} gene, the k^{th} column, $k \in J=\{1, 2, \dots, m\}$, represents the k^{th} condition and the cell $M[i, k]$ represents the expression level of the i^{th} gene under the k^{th} condition. A *bicluster* of M is a couple (I', J') such that $I' \subseteq I$ and $J' \subseteq J$.

The *biclustering problem* consists in extracting from a data matrix $M(I, J)$ a group of biclusters that maximize a given evaluation function. The biclustering problem is known to be NP-hard [10,22]. In the literature there are two main approaches for biclustering: the *systematic search* approach and the *stochastic search* or *metaheuristic* approach. Notice that most of these approaches are approximate methods.

The *systematic search* approach includes greedy algorithms [6,9,10,29], divide-and-conquer algorithms [17,26] and enumeration algorithms [4,20]. The *metaheuristic* approach includes neighbourhood-based algorithms [8], GRASP [12,13]

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 219–229, 2010.

and evolutionary algorithms [15,16,23]. A recent review of various biclustering algorithms for biological data analysis is provided in [3].

In this paper, we present a first adaptation of Iterative Local Search (ILS) to the biclustering problem. The resulting algorithm, called BILS, integrates several original features. BILS employs a new evaluation function for the assessment of biclusters. In BILS, we introduce a dedicated neighborhood relation which allows the search to improve gradually the quality of bicluters. To allow the search to escape from local optima, BILS uses a randomized, yet guided perturbation strategy.

To assess the performance of BILS, we applied BILS to the well-known yeast cell-cycle dataset and validated the extracted biclusters using external biological information by determining the functionality of the genes of the biclusters from the Gene Ontology database [2] using GOTermFinder tool[1]. Genes belonging to our biclusters were found to be significantly enriched with GO terms with very small p-values. We also use the web tool FuncAssociate [7] to compute the adjusted p-values. Our biclusters were found to be statistically significant with adjusted p-values < 0.001. We also compared our algorithm with two popular biclustering algorithms of Cheng and Church (CC) [10] and OPSM [6].

The remainder of the paper is organized as follows: In section 2, we describe our new biclustering algorithm. In section 3, we carry out an experimental study of BILS and assess its results using the above cited web-tools. Finally, in the last section, we present our conclusion and perspective.

2 The BILS Algorithm

2.1 Iterated Local Search

Iterated Local Search can be described by a simple computing schema [19]. A fundamental principle of ILS is to exploit the tradeoff between intensification and diversification. Intensification focuses on optimizing the objective function as far as possible within a limited search region while diversification aims to drive the search to explore new promising regions of the search space. The diversification mechanism of ILS–perturbation operator–has two aims: one is to jump out of the local optimum trap; the other is to lead the search procedure to a new promising region.

From the operational point of view, An ILS algorithm starts with an initial solution and performs local search until a local optimum is found. Then, the current local optimum solution is perturbed and another round of local search is performed with the perturbed solution.

Our BILS algorithm follows this general ILS schema. It uses a Hill-climbing (HC) algorithm as its local search procedure. In the rest of this section, we explain the main ingredients of this HC algorithm as well as the perturbation-based diversification strategy.

[1] http://db.yeastgenome.org/cgi-bin/GO/goTermFinder

2.2 Preprocessing Step: Construction of the Behavior Matrix

Prior to the search step using ILS, our method first uses a preprocessing step to transform the input data matrix M to a *Behavior Matrix* M'. This preprocessing step aims to highlight the trajectory patterns of genes. Indeed, according to [21,24,27], in microarray data analysis, genes are considered to be in the same cluster if their trajectory patterns of expression levels are similar across a set of conditions. In our case, each column of M' represents the trajectory of genes between a pair of conditions in the data matrix M. The whole M' matrix provides useful information for the identification of related biclusters and the definition of a meaningful neighborhood and perturbation strategy.

Formally, the Behavior Matrix M' is constructed progressively by merging a pair of columns (conditions) from the input data matrix M. Since M has n rows and m columns, there is $m(m-1)/2$ distinct combinations between columns, represented by J''. So, M' has n rows and $m(m-1)/2$ columns. M' is defined as follows:

$$M'[i,l] = \begin{cases} 1 & \text{if } M[i,k] < M[i,q] \\ -1 & \text{if } M[i,k] > M[i,q] \\ 0 & \text{if } M[i,k] = M[i,q] \end{cases} \tag{1}$$

with $i \in [1..n]$, $l \in [1..J'']$, $k \in [1..m-1]$, $q \in [1..m]$ and $q > k+1$.

Using M', we can observe the behavior of each gene through all the combined conditions. In our case, the combination of all conditions gives useful information since a bicluster may contains a subset of non contiguous conditions.

2.3 Initial Solutions and Basic Search Process

Given the Behavior Matrix M', our BILS algorithm explores iteratively different biclusters. To do this, BILS needs an initial bicluster (call it s_0) as its starting point. This initial bicluster can be provided by any means. For instance, this can be done randomly with a risk of starting with an initial solution of bad quality. A more interesting strategy is to employ a fast greedy algorithm to obtain rapidly a bicluster of reasonable quality. We use this strategy in this work and adopt two well-known algorithms: one is presented by Cheng and Church [10] and the other is called OPSM which is introduced in [6].

Starting from this initial solution, BILS will try to find iteratively biclusters of better and better quality. Basically, the improvement is realized by removing a "bad" genes from the current bicluster and adding one or more other "better" genes. Each application of this dual drop/add operation generates a new bicluster from the current bicluster. The way of identifying the possible genes to drop and to add defines the so-called neighborhood which is explained in detail in section 2.6.

2.4 Solution Representation and Search Space

A candidate solution is simply a bicluster and represented by $s = (I', J')$. As explained in the next section, our algorithm explores different biclusters with variable number of genes and a fixed number of conditions. The search space is thus determined by the number k of genes in the initial bicluster and has size of 2^g where $g = n - k$.

2.5 Evaluation Function

For a given solution (bicluster), its quality is assessed by an evaluation function. One of the most popular evaluation functions in the literature is called *Mean Squared Residue* (MSR) [10]. MSR has been used by several biclustering algorithms [9,13,23]. Yet MSR is known to be deficient to assess correctly the quality of certain types of biclusters like multiplicative models [1,25,29,9]. Recently, Teng and Chan [29] proposed another function for bicluster evaluation called Average Correlation Value (ACV). However, the performance of ACV is known to be sensitive to errors [9]. Both MSR and ACV are designed to be applied to the initial data matrix M. In our case, since M is preprocessed to obtain M', the above mentioned evaluation functions cannot be applied. For these reasons, we propose a new evaluation function $\mathbb{S}$ to evaluate a bicluster.

Given a candidate solution (a bicluster) $s = (I', J')$, the quality of s is assessed *via* the following score function $\mathbb{S}(s)$:

$$\mathbb{S}(s) = \frac{\sum_{i \in I'} \sum_{j \in I', j > i+1} \mathcal{F}_{ij}(g_i, g_j)}{|I'|(|I'| - 1)/2} \tag{2}$$

with $\mathcal{F}_{ij}(.,.)$ being defined by:

$$\mathcal{F}_{ij}(g_i, g_j) = \frac{\sum_{l \in J''_{s_0}} T(M'[i,l] = M'[j,l])}{|J''_{s_0}|} \tag{3}$$

where

- $T(Func)$ is true, if and only if $Func$ is true, and $T(Func)$ is false otherwise.
- $i \in I'$, $j \in I'$ and $i \neq j$, when $\mathcal{F}$ is used by $\mathbb{S}$ and, $i \in I$, $j \in I$ and $i \neq j$ otherwise.
- $|J''_{s_0}|$ is the cardinality of the subset of conditions in M' obtained from s_0,
- $0 \leq \mathcal{F}_{ij}(g_i, g_j) \leq 1$.

In fact, each $\mathcal{F}$ score assesses the quality of a pair of genes (g_i, g_j) under the subset of conditions of s. A high (resp. low) $\mathcal{F}_{ij}(g_i, g_j)$ value, *close* to 1 (resp. *close* to 0), indicates that the genes (g_i, g_j) (under the given conditions) are strongly (resp. weakly) correlated.

Given two pairs of genes (g_i, g_j) and (g'_i, g'_j), it is then possible to compare them: (g_i, g_j) is better than (g'_i, g'_j), when $\mathcal{F}_{ij}(g_i, g_j) > \mathcal{F}_{ij}(g'_i, g'_j)$.

Furthermore, $\mathbb{S}(s)$ is an average of $\mathcal{F}_{ij}(g_i, g_j)$ for each pair of genes in s. So, $0 \leq \mathbb{S}(s) \leq 1$. As $\mathcal{F}_{ij}(g_i, g_j)$, a high (resp. low) $\mathbb{S}(s)$ value, *close* to 1 (resp. *close* to 0), indicates that the solution s is strongly (resp. weakly) correlated.

Now given two candidate solutions s and s', s is better than s' if $\mathbb{S}(s) > \mathbb{S}(s')$.

2.6 Move and Neighborhood

One of the most important features of a local search algorithm is its neighborhood. In a local search algorithm, applying a move operator mv to a candidate solution s leads to a new solution s', denoted by $s' = s \oplus mv$. Let $\Gamma(s)$ be the set of all possible moves which can be applied to s, then the neighborhood $N(s)$ of s is defined by: $N(s) = \{s \oplus mv | mv \in \Gamma(s)\}$.

In our case, the move is based on the drop/add operation which removes a gene $\{g_i | i \in I'\}$ from the solution s and add another gene $\{g_v | v \notin I'\}$ or several other genes $\{g_v, \ldots, g_w | v \notin I', \ldots, w \notin I'\}$ to s.

The move operator can be defined as follows. Let $s = (I', J')$ be a solution and let $\lambda \in [0..1]$ be a fixed quality *threshold* (See Section 2.5 for quality evaluation). For each $i \in I', j \in I', r \in I'$ and $i \neq j \neq r$, we first choose a pair of genes (g_i, g_j) such that $\mathcal{F}_{ij}(g_i, g_j) < \lambda$. Such a pair of genes shows that they contributes negatively to the quality of the bicluster when they are associated. Now we look for another pair of genes (g_j, g_r) satisfying $\mathcal{F}_{jr}(g_j, g_r) \geq \lambda$. By this choice, we know that g_j contributes positively to the quality of the bicluster when it is associated with g_r. Notice that for both choices, ties are broken at random in order to introduce some diversification in the move operator.

Finally, we remove g_i which is a bad gene among the genes belonging to I' and we add all the genes $\{g_v, \ldots, g_w | v \notin I', \ldots, w \notin I'\}$ such that the values $\mathcal{F}_{rv}(g_r, g_v), \ldots, \mathcal{F}_{rw}(g_r, g_w)$ are higher than or equal to λ. Such an operator clearly help improve the quality of a bicluster, but also maximize the bicluster size [14,23].

Applying the move operator to a solution s leads to a new bicluster s', called neighboring solution or simply neighbor. For a given bicluster s, all possible neighbors define its neighborhood $N(s)$. It is clear that a neighboring solution s' has at least as many genes as in the original solution s.

2.7 The General BILS Procedure

The general BILS procedure is given in Algorithm 1. Starting from an initial solution (call it current solution s, see section 2.3), our BILS algorithm uses the Hill-climbing strategy to explore the above neighborhood. At each iteration, we move to an improving neighboring solution $s' \in N(s)$ according to the evaluation function $\mathbb{S}(s)$. This Hill-climbing based intensification phase stops when no improving neighbor can be found in the neighborhood. So, the last solution is the best solution found and corresponds to a local optimum. At this point, BILS triggers a diversification phase by perturbing the best solution to generate a new starting point for the next round of the search.

Our perturbation operator changes the best local optimum by deleting randomly 10% of genes of the best solution and adding 10% of genes among the best genes that are not included in the best solution. This perturbed solution is used by BILS as its new starting point.

The whole BILS algorithms stops when the best bicluster reaches a fixed quality or when the best solution found is not updated for a fixed number of perturbations.

Algorithm 1. General BILS Procedure

```
Input: An initial bicluster $s_0$, quality threshold $\lambda$
Output: The best bicluster
Create the Behaviour Matrix $M'$
Compute $\mathcal{F}$ for all pairs of genes to create $\Gamma(s_0)$
$s = s_0$ // current solution
repeat
    repeat
        Choose a pair of genes $(g_i, g_j)$ belonging to $s$ such that $\mathcal{F}_{ij}(g_i, g_j) < \lambda$
        Choose a pair of genes $(g_j, g_r)$ belonging to $s$ such that $\mathcal{F}_{jr}(g_j, g_r) \geq \lambda$
        Identify all genes $g_v$, $v \notin I'$ such that $\mathcal{F}_{rv}(g_r, g_v) \geq \lambda$
        Generate neighbor $s'$ by dropping $g_i$ from $s$ and adding all $g_v$
        if $(\mathbb{S}(s') \geq \mathbb{S}(s))$ then $s = s'$
        endif
    until (no improving neighbor can be found in $N(s)$)
    Generate a new solution $s$ by perturbing randomly 10% of the best solution
until (stop condition is verified)
Return $s$
```

3 Experimental Results

3.1 Dataset and Experimental Protocol

In order to analyze the effectiveness of the proposed algorithm, we used the well-known yeast cell-cycle microarray dataset. The yeast cell-cycle dataset is described in [28]. It is processed in [10] and publicly available from [11]. It contains the expression profiles of more than 6000 yeast genes measured at 17 conditions over two complete cell cycles. In our experiments we use 2884 genes selected by [10].

The obtained results have been compared with two popular biclustering algorithms: the one proposed by Cheng and Church (CC) [10] and OPSM described in [6]. For these reference algorithms, we have used *Biclustering Analysis Toolbox* (BicAT) which is a recent software platform for clustering-based data analysis that integrates these biclustering algorithms [5].

For this experiment, the λ threshold of BILS is experimentally set to 0.7. In fact, for each experiment ten values are tested between 0.1 and 1 with a stepwise of 0.1. With $\lambda = 0.7$, we have obtained the lowest p-values. The threshold δ of CC is selected as 300 like used in [10] and the default parameter setting is used for OPSM. With these algorithms, we have obtained 10 biclusters for CC and 14 biclusters for OPSM. Post-filtering was applied in order to eliminate insignificant biclusters like Cheng *et al.* [9]. This led to 8 biclusters CC and for 10 biclusters for OPSM. These biclusters are used as initial solutions for BILS and we compare the outputs of BILS with these initial biclusters.

The two web tools Funcassociate [7] and GoTermFinder[2] are used to evaluate statistically and biologically the biclusters.

[2] `http://db.yeastgenome.org/cgi-bin/GO/goTermFinder`

Our algorithm is run on a PC with 3.00GHz CPU and 3.25Gb RAM. Computing time is not reported, but let us mention that to improve one bicluster it takes between 3 and 11 minutes.

3.2 Statistical and Biological Significance Evaluation

Statistical significance of the biclusters is obtained by using the Funcassociate [7] web tool to compute the p-values and the adjusted p-values.

First, we asses the quality of the group of 18 biclusters obtained by BILS when it is applied to the 8 initial biclusters provided by CC and 10 initial biclusters given by OPSM. Funcassociate is used to compute the adjusted p-values of each of our 18 biclusters, leading always to an adjusted p-values < 0.001. This indicates that all these biclusters are statistically significant.

Now we turn our attention to the interpretation of results using the p-values. In fact, the p-values show how well they match with the known gene annotation. The closer the p-value is to zero, the more significant is the association of the particular Gene Ontology (GO) with the group of genes. For this purpose, we decide to examine for each algorithm only two biclusters: the bicluster having the maximum p-value and the one having the minimum p-value. Let B_xx_{MaxP} (resp. B_xx_{MinP}) denote these biclusters for algorithm $xx =$ CC or $xx =$ OPSM.

Table 1 summarizes the largest (column 2) and the smallest (column 3) p-values of the eight biclusters obtained from CC and the ten biclusters obtained from OPSM. The obtained biclusters from these algorithms with largest/smallest p-values are improved with BILS (row 3 for CC and 5 for OPSM). For instance, the element 0.000010 at row 2 and column 2 is the p-value of the bicluster B_CC_{MaxP} of CC while the element 2.220e-17 at row 3 and column 2 is the p-value of the improved bicluster B_CC_{MaxP} by BILS.

From the table, we see that BILS successfully improves the biclusters of CC and OPSM. In fact, both the maximum and minimum p-values of BILS are always better than those of CC and OPSM. This demonstrates that BILS is able to replace bad genes of the candidate solution by good genes by applying our move operator. Thus we can say that the biclusters of BILS are more statistically significant than those of CC and OPSM.

Table 1. P-values of the genes of the biclusters for BILS, CC and OPSM

Algorithms	Maximum p-value	Minimum p-value
CC	0.000010	4.096e-40
BILS	2.220e-17	2.860e-70
OPSM	0.0000012	1.587e-13
BILS	1.156e-10	4.865e-24

In addition to the above statistical significance validation, we also apply the GoTermFinder web tool on the biclusters used at the Table 1 to evaluate their biological significance, i.e., to show significant enrichment with respect to a specific GO annotation, in terms of associated biological processes, molecular functions and cellular components respectively compared to CC and OPSM.

Table 2. Most significant shared GO terms (biological process, molecular function, cellular component) of CC and BILS for two biclusters on yeast cell-cycle dataset

Algorithms	Biological Process	Molecular function	Cellular component
CC (B_CC_{MaxP})	unknown	unknown	Cytoplasm (0.00932)
BILS$_{CC}$: improved B_CC_{MaxP} by BILS	Maturation of SSU-rRNA (4.54e-05) Maturation of SSU-rRNA from tricistronic rRNA transcript(SSU-rRNA, 5.8S rRNA, LSU-rRNA) (0.00088) Cell cycle (0.00107)	structural constituent of ribosome (4.14e-17) Structural molecule activity (1.97e-15)	cytosolic ribosome (2.94e-21) ribosomal subunit (4.27e-17) cytosolic part (2.04e-16)
CC (B_CC_{MinP})	translation (8.33e-23) cellular protein metabolic process (3.17e-10) gene expression (6.48e-10)	structural constituent of ribosome (1.03e-36) structural molecule activity (3.91e-28) helicase activity (0.00021)	cytosolic ribosome (7.83e-42) ribosome (3.80e-36) cytosolic part (1.82e-35)
BILS$_{CC}$: improved B_CC_{MinP} by BILS	translation (2.86e-35) cellular protein metabolic process (2.59e-16) cellular macromolecule biosynthetic process (1.74e-15)	structural constituent of ribosome (2.50e-70) Structural molecule activity (6.06e-54) translation factor activity, nucleic acid binding (0.00445)	cytosolic ribosome (1.05e-76) ribosomal subunit (1.08e-68) cytosolic part (1.01e-66)

For this, Table 2 and 3 describe the top GO terms of the three categories with the lowest p-values. The value within parentheses after each GO term, e.g., Table 2 second column third line, such as (4.54e-05) indicates the statistical significance which is provided by the p-value. We observe that BILS can obtain improved biclusters not only in terms of p-values, i.e., quality of biclusters, but also in terms of GO annotation. For example Table 2 (resp. Table 3) shows that CC (resp. OPSM) can not identify any biological process and molecular functions (resp. biological process and cellular component) for the bicluster B_CC_{MaxP} (resp. B_OPSM_{MinP}). However, BILS can produce biclusters with all categories, i.e., biological processes, molecular functions and cellular components. This shows that our algorithm is able to identify biological significant biclusters.

Table 3. Most significant shared GO terms (biological process, molecular function, cellular component) of OPSM and BILS for two biclusters on yeast cell-cycle dataset

Algorithms	Biological Process	Molecular function	Cellular component
OPSM (B_OPSM_{MaxP})	sister chromatid segregation (0.00337) chromosome segregation (0.00478) microtubule-based process (0.00588)	unknown	spindle (0.00196) microtubule cytoskeleton (0.00295) chromosomal part (0.00991)
BILS$_{OPSM}$: improved B_OPSM_{MaxP} by BILS	cellular component organization (1.71e-07) nucleic acid metabolic process (1.72e-06) cellular nitrogen compound metabolic process (7.88e-06)	structural constituent of cytoskeleton (0.00099) RNA polymerase II transcription factor (0.00640)	nucleus (3.83e-12) nuclear part (3.91e-09) chromosomal (2.26e-08)
OPSM (B_OPSM_{MinP})	unknown	oxidoreductase activity (6.78e-06) oxidoreductase activity, acting on CH-OH group of donors (0.00075) oxidoreductase activity, acting on peroxidase as acceptor (0.00078)	unknown
BILS$_{OPSM}$: improved B_OPSM_{MinP} by BILS	response to stimulus (0.00092) response to stress (0.00454)	structural constituent of ribosome (9.19e-24) structural molecule activity (3.78e-12) oxidoreductase activity (2.36e-05)	cytosolic ribosome (1.09e-23) ribosomal subunit (3.28e-23) cytosolic part (7.35e-22)

4 Conclusion and Future Work

In this paper, we have presented a new biclustering algorithm using Iterative Local Search (BILS). BILS combines a dedicated Hill-climbing based local search procedure and a perturbation strategy. For the intensification purpose, BILS employs a new evaluation function and a dedicated neighborhood relation. We have tested and assessed our algorithm on the yeast cell-cycle dataset. The experimental results show that the BILS algorithm can successfully improve all biclusters of CC and OPSM according to statistical and biological evaluation criteria.

The work reported in this paper correspond in fact to an ongoing study. Several improvements to the proposed work can be envisaged. One immediate possibility would be to study alternative neighborhoods to introduce more biological knowledge to provide more effective guidance of the local search process. Another natural extension would be to reinforce the basic local search procedure by more powerful metaheuristics such as Tabu Search. Moreover, BILS explores the space of biclusters by changing only the subset of genes of a bicluster without changing the conditions of the initial bicluster. It is natural to design similar strategies to optimize the subset of conditions of a bicluster or eventually to optimize simultaneously both the set of genes and conditions. Finally, another possible experimentation is to assess the algorithm on a synthetic data.

Acknowledgements

We are grateful for comments by the referees that have improved the paper. This research was partially supported by the region Pays de la Loire (France) with its ‘Bioinformatics Program’ (2007-2010).

References

1. Aguilar-Ruiz, J.S.: Shifting and scaling patterns from gene expression data. Bioinformatics 21, 3840–3845 (2005)
2. Ashburner, M., Ball, C.A., Blake, J.A., Bolstein, D., Butler, H., Cherry, M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubinand, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. the gene ontology consortium. Nature Genetics 25, 25–29 (2000)
3. Ayadi, W., Elloumi, M.: Biclustering of Microarray Data. In: Algorithms in Computational Molecular Biology: Techniques, Approaches and Applications. John Wiley & Sons Inc., Chichester (to appear 2010)
4. Ayadi, W., Elloumi, M., Hao, J.K.: A biclustering algorithm based on a bicluster enumeration tree: Application to dna microarray data. BioData Mining 2, 9 (2009)
5. Barkow, S., Bleuler, S., Prelic, A., Zimmermann, P., Zitzler, E.: Bicat: a biclustering analysis toolbox. Bioinformatics 22(10), 1282–1283 (2006)
6. Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z.: Discovering local structure in gene expression data: the order-preserving submatrix problem. In: RECOMB ’02: Proceedings of the sixth annual international conference on Computational biology, pp. 49–57. ACM, New York (2002)
7. Berriz, G.F., Beaver, J.E., Cenik, C., Tasan, M., Roth, F.P.: Next generation software for functional trend analysis. Bioinformatics 25(22), 3043–3044 (2009)
8. Bryan, K., Cunningham, P., Bolshakova, N.: Application of simulated annealing to the biclustering of gene expression data. IEEE Transactions on Information Technology on Biomedicine 10(3), 519–525 (2006)
9. Cheng, K.O., Law, N.F., Siu, W.C., Liew, A.W.: Identification of coherent patterns in gene expression data using an efficient biclustering algorithm and parallel coordinate visualization. BMC Bioinformatics 9(210) (2008)
10. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, pp. 93–103. AAAI Press, Menlo Park (2000)
11. Cheng, Y., Church, G.M.: Biclustering of expression data. Technical report (supplementary information) (2006)
12. Das, S., Idicula, S.M.: Application of reactive grasp to the biclustering of gene expression data. In: ISB ’10: Proceedings of the International Symposium on Biocomputing, pp. 1–8. ACM, New York (2010)

13. Dharan, A., Nair, A.S.: Biclustering of gene expression data using reactive greedy randomized adaptive search procedure. BMC Bioinformatics 10(suppl. 1), S27 (2009)
14. Divina, F., Aguilar-Ruiz, J.S.: Biclustering of expression data with evolutionary computation. IEEE Transactions on Knowledge and Data Engineering 18(5), 590–602 (2006)
15. Divina, F., Aguilar-Ruiz, J.S.: A multi-objective approach to discover biclusters in microarray data. In: GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation, pp. 385–392. ACM, New York (2007)
16. Gallo, C.A., Carballido, J.A., Ponzoni, I.: Microarray biclustering: A novel memetic approach based on the pisa platform. In: Pizzuti, C., Ritchie, M.D., Giacobini, M. (eds.) EvoBIO 2009. LNCS, vol. 5483, pp. 44–55. Springer, Heidelberg (2009)
17. Hartigan, J.A.: Direct clustering of a data matrix. Journal of the American Statistical Association 67(337), 123–129 (1972)
18. Hoos, H., Stutzle, T.: Stochastic Local Search: Foundations and Applications. Morgan Kaufmann, San Francisco (2004)
19. Lourenco, H.R., Martin, O., Stützle, T.: Iterated local search. In: Glover, F., Kochenberger, G. (eds.) Handbook of Meta-heuristics, pp. 321–353. Springer, Heidelberg (2003)
20. Liu, J., Wang, W.: Op-cluster: Clustering by tendency in high dimensional space. In: IEEE International Conference on Data Mining, pp. 187–194 (2003)
21. Luan, Y., Li, H.: Clustering of time-course gene expression data using a mixed-effects model with B-splines. Bioinformatics 19(4), 474–482 (2003)
22. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) 1(1), 24–45 (2004)
23. Mitra, S., Banka, H.: Multi-objective evolutionary biclustering of gene expression data. Pattern Recogn. 39(12), 2464–2477 (2006)
24. Peddada, S.D., Lobenhofer, E.K., Li, L., Afshari, C.A., Weinberg, C.R., Umbach, D.M.: Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference. Bioinformatics 19(7), 834–841 (2003)
25. Pontes, B., Divina, F., Giráldez, R., Aguilar-Ruiz, J.S.: Virtual error: A new measure for evolutionary biclustering. In: Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics, pp. 217–226 (2007)
26. Prelic, A., Bleuler, S., Zimmermann, P., Buhlmann, P., Gruissem, W., Hennig, L., Thiele, L., Zitzler, E.: A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22(9), 1122–1129 (2006)
27. Schliep, A., Schonhuth, A., Steinhoff, C.: Using hidden Markov models to analyze gene expression time course data. Bioinformatics 19(Suppl. 1), i255–i263 (2003)
28. Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., Church, G.M.: Systematic determination of genetic network architecture. Nature Genetics 22, 281–285 (1999)
29. Teng, L., Chan, L.: Discovering biclusters by iteratively sorting with weighted correlation coefficient in gene expression data. J. Signal Process. Syst. 50(3), 267–280 (2008)

Biologically-aware Latent Dirichlet Allocation (BaLDA) for the Classification of Expression Microarray

Alessandro Perina[1], Pietro Lovato[1],
Vittorio Murino[1,2], and Manuele Bicego[1,2,⋆]

[1] University of Verona, Verona, Italy
[2] Italian Institute of Technology (IIT), Genova, Italy
Tel.: +39 045 8027072, Fax: +39 045 8027968
manuele.bicego@univr.it

Abstract. Topic models have recently shown to be really useful tools for the analysis of microarray experiments. In particular they have been successfully applied to gene clustering and, very recently, also to samples classification. In this latter case, nevertheless, the basic assumption of functional independence between genes is limiting, since many other a priori information about genes' interactions may be available (co-regulation, spatial proximity or other a priori knowledge). In this paper a novel topic model is proposed, which enriches and extends the Latent Dirichlet Allocation (LDA) model by integrating such dependencies, encoded in a categorization of genes. The proposed topic model is used to derive a highly informative and discriminant representation for microarray experiments. Its usefulness, in comparison with standard topic models, has been demonstrated in two different classification tests.

1 Introduction

Microarrays represent a widely employed tool in molecular biology and genetics, which have produced an enormous amount of data to be processed to infer knowledge. Computational methodologies may be very useful in such analysis: among others, clear examples are tools aiding the microarray probe design, image processing-based techniques for the quantification of the spots, segmentation of spots/background, grid matching, noise suppression [5], methodologies for classification or clustering [22]. In this paper we focus on this last class of problems, and in particular on the samples classification task. In this context, many approaches have been presented in the literature in the past, each one characterized by different features, like computational complexity, effectiveness, interpretability, optimization criterion and others – for a review see e.g. [13,21].

In particular, very recently, a class of approaches have shown to be useful and discriminant in this context: the so called *topic* or *latent models* – the two most famous examples being the Probabilistic Latent Semantic Analysis (PLSA – [10])

[⋆] Corresponding author.

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 230–241, 2010.

and the Latent Dirichlet Allocation (LDA – [3]). These powerful approaches have originally been introduced in the text analysis community for unsupervised topic discovery in a corpus of documents, in order to correlate the presence of a word to the particular topic discussed; the whole corpus of documents can then be described in terms of these topics. These techniques have also been largely applied in the computer vision community [4].

One of the main characteristics of this class of approaches is represented by their interpretability [7]: they can model a dataset in terms of hidden topics (or processes), which can reflect underlying and meaningful structures in the problem. This characteristic may be extremely useful in bioinformatics, where interpretability of methods and results is crucial. Topic models have already been applied in the context of expression microarray analysis: a tailored version of LDA (called Latent Process Decomposition – LPD), explicitly modelling expression levels, has been proposed in [19], with the aim of clustering expression microarray data; moreover, an application of topic models to biclustering has been recently proposed in [1].

A somehow unexplored scenario is represented by the application of such models in the classification context – a preliminary evaluation of standard topic models have been recently proposed in [2]. Even if supported by very promising results, a clear drawback is represented by the underlying basic assumption that each gene expression is independently generated given its corresponding latent topic.

In this paper a novel topic model is proposed, which we call BaLDA (Biologically-aware Latent Dirichlet Allocation), which starts from the Latent Process Decomposition [19], introduced in the context of clustering, and defines a new model able to take into account the given dependence between genes. This dependence is introduced in the graphical model through a variable, modeling a categorization of genes (namely a subdivision of genes in groups), which can be inferred by a priori knowledge on the genes of the analyzed problem. As a further refinement, a better modelling of the expression level is achieved by substituting the Gaussian pdf – present in the LPD – with a more descriptive Mixture of Gaussians.

We will show the usefulness of BaLDA in two classification experiments, assessing the impact of the different introduced modifications; a comparison with the LPD topic models and state of the art methods demonstrates the competitiveness of the proposed approach.

The rest of the paper is organized as follows: in Sec. 2 technical preliminaries about topic models are given. In Sec. 3 the model, together with learning/inference mechanism presented. An exhaustive experimental section is presented in Sec. 4, and, finally, in Sec. 5, we draw some conclusions.

2 Background

In this section the background concepts are reviewed. In particular, after introducing the general ideas underlying the family of topic models, we will present

Laten Dirichlet Allocation using the terminology and the notation of the document analysis context. Then we will briefly review how these models have been applied to the microarray scenario.

2.1 Topic Models

Topic models were introduced in the linguistic scenario, in order to describe and model documents. The basic idea underlying these methods is that each document is characterized by the presence of several topics (e.g. sport, finance, politics), which induce the presence of some particular words. From a probabilistic point of view, the document may be seen as a mixture of topics, each one providing a probability distribution over words.

A variety of probabilistic topic models have been used to analyze the content of documents and the meaning of words. In the following section we will briefly present the LDA model, mainly to set up notations used in the remainder of the paper.

2.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) was first introduced by Blei in [3]. In the LDA model, words are the only observable variables and they implicitly reflect a latent structure, i.e., the set of K topics used to generate the document. Generally speaking, given a set of documents, the latent topic structure lies in the set of words itself. In generating the document, for each word-position a topic is sampled and, conditioned from the topic, a word is selected. Each topic is chosen on the basis of the random variable θ that is sampled for convenience from a Dirichlet distribution $p(\theta|\alpha)$ where α is a hyperparameter. The topic z conditioned on θ and the word w conditioned on the topic and on β are sampled from multinomial distributions $p(z_n|\theta)$ and $p(w_n|z_n,\beta)$ respectively. β represents the word distribution over the topics. Given the parameters α and β, the joint distribution of a topic mixture θ, a set of N topics z_n, and a set of N words w_n that compose the document is given by

$$p(\theta,\mathbf{z},\mathbf{w}|\alpha,\beta) = p(\theta|\alpha) \cdot \prod_{n=1}^{N} p(z_n|\theta) \cdot p(w_n|z_n,\beta) \tag{1}$$

where $p(z_n = i|\theta)$ is simply θ_i for the unique i such that $z_n^i = 1$. Integrating over θ and summing over z, we obtain the probability of a document.

2.3 Topics Models in Bioinformatics

The representation provided by topic models has one clear advantage: each topic is individually interpretable, providing a probability distribution over words that picks out a coherent cluster of correlated terms, see for example [6,2,19]. This may be really advantageous in the expression microarray context, since the final goal is to provide knowledge about biological systems, and discover possible hidden correlations. In particular there is a straightforward analogy between the pairs word-document and gene-sample: the expression level of a gene in a sample may be easily interpreted as the level of the presence of a word in a document (the higher the level the more present/expressed the word/gene is). In this sense, a particular topic model assumes that microarray data (represented as the gene-expression matrix) arises from a mixture of topics, whose number is fixed; changing the topic allows different subsets of genes to be prominent.

A possible problem which may arise is that expression microarray data is described with a matrix of real numbers, not as a non-negative integer matrix. This problem has been solved in [19] by modifying the standard LDA via the introduction of Gaussian distributions in place of word multinomial distributions β; this results in a novel and efficient probabilistic model called Latent Process Decomposition (LPD), where LDA topics are called "processes". The model has been successfully applied to clustering. Some modifications of the LPD model have been recently introduced: in particular, an optimized training version can be found in [23]; moreover, in [15], the LPD has been equipped with learned hyperpriors on the gaussian word-topic distributions. A method for maximizing lower bounds by re-estimating hyperparameters leaded to more accurate clustering results.

A somehow unexplored scenario is represented by the application of such models in the classification context; only very recently PLSA and LDA have been employed to classify expression microarray samples, with really promising results [2]. In particular, in [2], the original topic models [3,10] have not been changed; instead the gene expression matrix has been transformed, by a proper scaling and shifting, to a positive integer valued matrix, thus interpretable as a count matrix in the original LDA-PLSA formulation. Despite the method lacks biological motivations, it yielded very good classification results.

3 Biologically-aware Latent Dirichlet Allocation (BaLDA)

The main contribution of this paper is the definition a novel topic model for the analysis of expression microarray data, which directly improves the one provided in [19]. This novel topic model has two clear advantages with respect to the Latent Process Decomposition (LPD), detailed in the following.

The first (and most important) advantage starts from the observation that the major drawback of the PLSA, LDA and LPD models is the assumption that each gene expression is independently generated given its corresponding latent topic. While such representation provides an efficient computational method, it lacks the power to describe the coherent expression of different genes in a subset of samples, this aspect being widely known in the biology. In the proposed approach we include a mechanism in the graphical model that permits to include a priori knowledge on the relation between genes. This a priori information is expressed in terms of a gene categorization, namely a subdivision of the genes in groups of related genes based on external information, like known co-regulation,

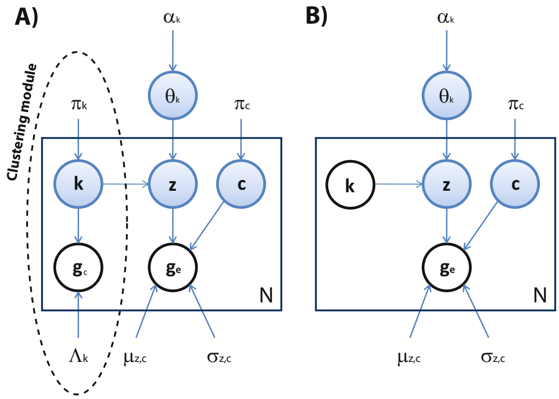


Fig. 1. A) Biologically-aware Latent Dirichlet Allocation Bayesian Network. Shaded/ Unshaded nodes are visible/hidden variables ($\mathbf{v}/\mathbf{h}$). The model parameters Ω are represented with a letter outside a node. B) A second version of the Biologically-aware Latent Dirichlet Allocation. The clustering result is fed into the model by means of the visible variable k.

spatial proximity or similarity of nucleotidic sequences to name a few. This categorization (i.e. clustering), which may be directly fed to the model, can be computed beforehand or can be simultaneously estimated while estimating the topic model.

The former option result in a straightforward modification of LDA; we add a visible variable k that influences the hidden topic variable z (see Fig. 1B). More interesting is the latter option, which permits LDA to deal with the uncertainty associated to the clustering. In this case (see Fig. 1A), the variable k is hidden, and depends on the visible variable g_c which represents the external information. These variables are modelled through a set of parameters which are learned simultaneously with the other parameters.

The second novelty of the proposed approach is related to the modelling of the word/topic distribution: in the original Latent Dirichlet Allocation, a word is generated by a multimodal distribution β, where $\beta_{w,z}$ represents the probability of finding the word w when the document is "speaking" about the topic z. In the LPD [19] the word-topic probability is modeled by a single gaussian, thus reflecting the continuous nature of the expression level, which is not captured with the original discrete formulation. Nevertheless, the monomodal nature of the Gaussian may not properly capture the possibly multimodal behavior of the gene-topic distribution: in particular, within a gene, a topic can be assigned to a single expression level. This limitation is removed in the proposed model, where the single Gaussian is replaced by a mixture of C Gaussians; which for large C, goes towards the multimodal spirit of the original multinomial β, still maintaining the appealing characteristic of modelling continuous expression levels.

3.1 BaLDA

The Bayesian network of Biologically-aware Latent Dirichlet Allocation (BaLDA) is depicted in Fig.1. The model is characterized by two observations, $g_{\mathbf{c}}$ and $g_{\mathbf{e}}$ (visible variables $\mathbf{v}$) which respectively govern the clustering and the topic submodules.

The variable k clusters the N genes in K components, while the parameters Λ_k represent the parameters of the particular probability density function chosen. For microarray expression are often used gaussians, t-distributions or factor analyzers [16]. We used gaussian clustering, so $\Lambda_k = \{\mu_k, \sigma_k\}$

$$p(g_{\mathbf{c},n}|k, \Lambda) = p(g_{\mathbf{c},n}|\Lambda_k) = \frac{1}{\sqrt{(2\pi)}\sigma} \cdot e^{\left(\frac{(g_{\mathbf{c},n}-\mu_k)^2}{-2\sigma_k^2}\right)} \tag{2}$$

The parameter π_k is a multinomial distribution that represents the prior on the cluster assignment.

Each n-th gene expression $g_{\mathbf{e},n}$ is assigned a topic $z_n = \{1 \ldots Z\}$ evaluating the gene-topic distribution and using a topic prior θ. We have that

$$p(g_{\mathbf{e},n}|z, \mu, \sigma) = \sum_{[c]} p(g_{\mathbf{e},n}|z, c, \mu, \sigma) = \sum_{[c]} \pi_{z,c,n} \cdot \frac{1}{\sqrt{(2\pi)}\sigma} \cdot e^{\left(\frac{(g_{\mathbf{e},n}-\mu_{z,c,n})^2}{-2\sigma_{z,c,n}^2}\right)} \tag{3}$$

where is now visible the mixture of Gaussians palette we introduced. With $[c]$ we indicate the values the variable c can assume. The prior on such topic assignment depends on the co-regulated genes (see the link $k \to z$ in the Bayesian network).

$$p(z = a|\theta, k) = \theta_{k,a} \tag{4}$$

where θ_k are multinomial distributions that represent the topic proportions used to generate each sample. Each distribution θ_k is governed by a Dirichlet prior $p(\theta_k|\alpha_k)$, where α is hyperparameter that represent the strength of a topic within a dataset.

$$p(\{\theta_k\}|\{\alpha_k\}) = \prod_{[k]} p(\theta_k|\alpha_k) = \prod_{[k]} \Big(\frac{1}{\mathcal{Z}(\alpha)} \prod_{z} \theta_{k,z}^{\alpha_k - 1} \Big) \tag{5}$$

Again the products are taken over the values of k and z and $\mathcal{Z}(\alpha)$ is Dirichlet distribution normalization constant.

At this point we can write the joint probability which describes the generative model as

$$p(g_{\mathbf{c}}, g_{\mathbf{e}}, c, k, z, \theta|\alpha, \mu, \sigma, \Lambda, \pi_c, \pi_k) = p(c|\pi_c) \cdot p(k|\pi_k) \cdot p(\theta|\alpha)$$
$$\prod_{n} \Big(p(g_{\mathbf{c},n}|k, \Lambda) \cdot p(g_{\mathbf{e},n}|c, z, \mu, \sigma) \cdot p(z_n|\theta) \Big)$$

where each conditional distribution has already been parameterized.

3.2 Inference and Learning

Under the model so far described, each t-th observation g^t is characterized by four hidden variables $\mathbf{h}^t = \{k^t, c^t, z^t, \theta_k^t\}$ which in turn are governed by the following parameters $\Omega = \{\Lambda_k, \pi_k, \mu_c, \sigma_c, \pi_c, \alpha\}$.

As in LDA, exact inference is intractable: we approach it using the variational inference [12]. We introduce a tunable distribution $q(\mathbf{h})$ over the hidden variables which defines the free energy $\mathcal{F}$

$$\mathcal{F} = \sum_t \Big(\sum_{\mathbf{h}} q(\mathbf{h}^t) \log \frac{q(\mathbf{h}^t)}{p(\mathbf{g}^t, \mathbf{h}^t | \Omega)} \Big) \tag{6}$$

We used the following form for the approximate posterior distribution, $q(\mathbf{h}^t) = q(\theta^t) \cdot \prod_n q(z_n^t, c_n^t) \cdot q(k_n^t)$ with $q(\theta_k^t)$ being a Dirac function centered at the optimal vectors $\hat{\theta}^t$. After plugging the approximate posterior and the joint distribution in the free energy formulation, we can iteratively decrease $\mathcal{F}$ with the Expectation-Maximization (EM) algorithm. The EM algorithm alternates in minimizing the free energy with respect to $q(\mathbf{h})$ (*E-Step*) and with respect to the model parameters Ω (*M-Step*). When updating q, the only constraint is that $\sum_{h_i} q(h^t) = 1$ for each hidden variable h and for each sample t. The update rules are simply obtained by setting the derivatives of $\mathcal{F}$ equal to zero and this reduces to the following formulas:

$$q(z_n^t = a, c_n^t = b) \propto \pi_b \cdot \mathcal{N}(g_{\mathbf{e},n}; \mu_{a,b,n}, \sigma_{a,b,n}) \cdot e^{\left(\sum_{[k_n]} q(k_n^t) \cdot \left(\Psi(\hat{\theta}_{b,a}) - \Psi(\sum_{[k]} \hat{\theta}_{k,b}) \right) \right)} \tag{7}$$

where Ψ is the derivative of the logΓ function, computable via Taylor approximation (for further details see [3]), and $\mathcal{N}$ is the normal probability function (see Eq.2). The remaining updates of the E-step are

$$\hat{\theta}_{b,a}^t \propto \alpha_{b,a} + \sum_n q(k_n^t = b) \cdot q(z_n^t = a) \tag{8}$$

$$q(k_n^t = k) \propto \pi_k \cdot \mathcal{N}(g_{\mathbf{t},n}; \mu_k, \sigma_k) \tag{9}$$

In the M-step the collected posterior distributions q are used to compute an estimate $\hat{\Omega}$ of the model parameters

$$\mu_{n,c,z} = \frac{\sum_t q(z_n = z) \cdot q(c_n^t = c) \cdot g_{\mathbf{e},n}^t}{\sum_t q(z_n = z) \cdot q(c_n^t = c)} \tag{10}$$

$$\sigma_{n,c,z}^2 = \frac{\sum_t q(z_n = z) \cdot q(c_n^t = c) \cdot (g_{\mathbf{e},n}^t - \mu_{n,c,z})^2}{\sum_t q(z_n = z) \cdot q(c_n^t = c)} \tag{11}$$

$$\pi_{c,z,n} = \sum_t q(c_n = c) \cdot q(z_n = z) \tag{12}$$

The appropriate update on topic proportions' priors α_k can be obtained using a gradient descend

$$\{\hat{\alpha}_{k,a}\} = \arg\max \sum_t (\alpha_{k,a} - 1) \log \theta_{k,a} \tag{13}$$

subject to the appropriate normalization constraint.

We omit the update formulas for μ_k, σ_k^2 and π_k which can be computed in a very similar fashion.

3.3 Expression Microarray Samples Classification

In general, topic models have been originally introduced for clustering sets of documents: given the dataset, models are trained and analyzed in order to find clusters. Nevertheless, recently, they have been also successfully employed in the classification scenario – see for example [4,2]. The main idea is to employ a hybrid generative-discriminative approach [11], which exploits the generative model to extract a set of features to be classified with a discriminative classifier. More in detail, the training phase is carried out by first learning the models on the training set. Then a set of features is extracted from each sample; the transformed training set is then used to train a classifier. In the testing phase, the same feature extraction process is applied to the test sample, resulting in a feature vector to be classified using the trained classifier. In our work we employed the scheme proposed in [4,2], i.e. we employ the mixture of topics θ^t as sample descriptor. This have been demonstrated to be really discriminant [4,2]. Another benefit of this representation is that we are reducing the dimensionality from the number of genes N to the number of topics K, with $K \ll N$ – thus providing a compact and more interpretable representation. Finally, we are describing samples with a multinomial distribution whose characteristics will be exploited by the particular chosen classifiers.

4 Experiments

The proposed classification scheme has been evaluated using two different datasets, both related to tumors. The first derives from a study of prostate cancer by Dhanasekaran et.al [20], and consists of 54 samples with 9984 features. Such samples are subdivided in different classes: 14 samples are labelled as benign prostatic hyperplasia (labelled BPH), 3 as normal adjacent prostate (NAP), 1 as normal adjacent tumor (NAT), 14 as localized prostate cancer (PCA), 1 prostatitis (PRO) and 20 as metastatic tumors (MET). The 6 classes can be divided in three macro-classes: non-cancer (BPH,NAP,PRO), cancer (NAT,PCA), metastatic tumor (MET). This dataset has been also employed by the authors of [19] in their study for LPD. The second dataset we employed contains the expressions of 90 brain tissues used to study central nervous system embryonal tumor [18]. Each sample is characterized by 5920 features. The 90 samples include 60 with medulloblastomas, 10 with malignant gliomas, 5 with AT/RTs, 5 with renal/extrarenal rhabdoid tumors, 6 with supratentorial PNETs, and 4 normal cerebellum (5 classes in total). As in many expression microarray analysis, a beneficial effect may be obtained by selecting a sub group of genes, in order to limit the dimensionality of the problem and to reduce the possible redundancy present in the dataset. Here, as in [19], we decided to perform the experiments filtering the genes by variance and keeping only the top 500 genes.

In all the experiments we set $g_{\mathbf{c}} = g_{\mathbf{e}}$, namely we clustered the genes by looking at their expression levels in all the samples. This choice of course does not exploit the full potentiality of the method, but it permits to already obtain promising results (see tables below). Currently we are planning to perform an experiment by fully exploiting the potentialities of the model, considering different information (like spatial proximity or sequence similarity). In all the experiments, Z, K, and C, representing the number of topics, the number of clusters and the number of components in the mixture of Gaussians, respectively, are set in the following way: Z was found by applying the hold out log likelihood procedure described in [19], K has been automatically determined using Affinity Propagation [9] and $C = 3$ has been set after several tests.

In order to capture the different contributions of the two innovations of the model, we also tested the model with *i)* the clustering module but with only one Gaussian per gene (C=1), *ii)* the model enriched by the mixture of Gaussians gene-topic distribution, without the clustering information (K=1). We will refer to these two versions as BaLDA v1 and BaLDA v2 respectively.

The extracted features have been classified using Support Vector Machines employing a variety of kernels. Beside the standard linear kernel (LI), the probabilistic nature of the extracted features has been exploited by the use of different kernels on measures – also called information theoretic kernels [14], which provide similarity between probabilistic distributions; we employed some recent kernels, like the Kullbach-Leibler (KL), the Jensen-Shannon (JS) and the Jeffries kernels (JE). Finally we report also results with the K- Nearest Neighbor rule, using an approach similar to [2].

The proposed model has been compared with [19,2]. Even if [19] was designed for clustering data, it can be straightforwardly adapted to the classification scenario, following exactly the same hybrid scheme we employed. In order to have a fair comparison, we used the authors' implementation. Moreover, for a given choice of (K, Z) in BaLDA, we trained two LPD models: one with the same number of topics $Z_{LPD} = Z$, and one with the same complexity $Z_{LPD} = K \cdot Z$; this permits to give to the LPD the same number of processes that we have in our model. It is important to notice that the optimal Z for LPD, found by applying to the hold out log likelihood procedure, has been used also for BaLDA. In fact it is not obvious that the optimal Z_{LPD} will be the optimal for BaLDA as well. Classification errors have been computed using 10-fold cross validation (with 40 repetitions). In order to augment the statistical significance of the results, the generative models have been trained 4 times and results averaged.

Results, for both datasets, are reported in Table 1 and 2, respectively. From the tables it is evident the improvement obtained with the BaLDA models. In particular, in all the provided experiments the full model is performing better than the original LPD model (except in one case), with very remarkable improvements in the first dataset, also employed in the original paper of [19]. Moreover, by comparing the results of BaLDA v1 and BaLDA v2, we can observe that the improvement introduced by clustering the genes is more relevant than the other; however the combination of the two eventually yielded the best

Table 1. Results obtained from Prostate Cancer Dataset. See the text for the kernel abbreviations. We tested [2] also using the information theoretic kernels reporting the accuracies for the best Z.

	Z	K	C	LI	KL	JS	JE	KNN
LPD [19]	3	n.a.	n.a.	65.41	66.04	68.55	68.55	77.70
LPD [19]	12	n.a.	n.a.	86.16	82.39	85.53	85.53	82.22
[2]	3	n.a.	n.a.	82.38	83.64	78.60	84.90	77.89
BaLDA v1	3	4	1	86.80	88.68	88.05	89.94	88.17
BaLDA v2	3	1	3	77.98	76.73	76.73	75.47	76.67
BaLDA	3	4	3	89.94	89.31	91.20	**91.20**	85.24

Table 2. Results obtained from Brain Tumor Dataset. On the bottom, we reported the best accuracies of three other state of the art methods.

	Z	K	C	LI	KL	JS	JE	KNN
LPD [19]	15	n.a.	n.a.	83.33	81.48	81.85	84.07	78.56
LPD [19]	90	n.a.	n.a.	66.67	66.67	66.67	66.67	82.11
BaLDA v1	15	6	1	85.56	85.56	88.15	88.52	82.74
BaLDA v2	15	1	3	76.67	84.08	76.67	80.37	76.48
BaLDA	15	6	3	85.19	85.19	87.87	**88.89**	81.15

Comparison with the state of the art

Method	Acc.	Method	Acc.	Method	Acc.
[17]	86.50	[8]	86.20	[2]	84.1

result. Considering the classifiers, it is not clear which is the best combination of kernels and classifiers – this depending on the given dataset and on the given generative model. As a general comment, it can be said that information theoretic kernels are working better than the linear one, so confirming the intuition that exploiting the probabilistic nature of the features may be useful.

A final comment regards the interpretability of the method. Figure 2 describes topic proportions of the different models. We can observe that the topics can capture the different classes of the problem (with our model producing a qualitative better result – for more comments see the caption of the figure). This appealing interpretability of the topic models has been recently exploited in a biclustering scenario (see [1]).

5 Conclusions

In this paper we proposed a novel topic model, which enriches and extends the Latent Dirichlet Allocation (LDA) model by integrating genes' dependencies, encoded in a categorization of genes which better models the gene-topic distribution, leading to better classification of samples. The proposed model, called

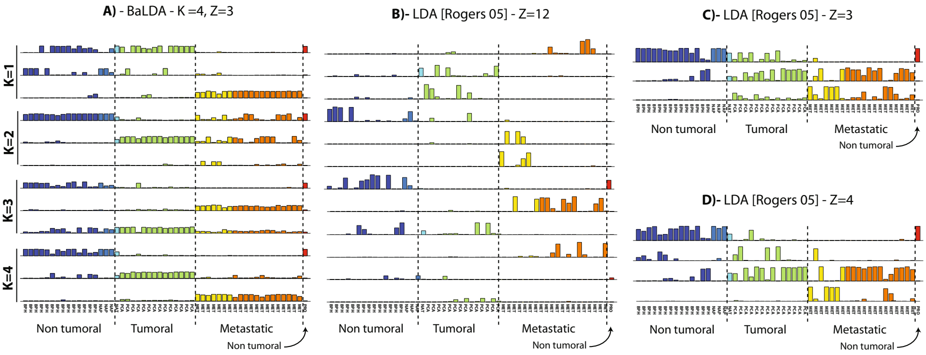


Fig. 2. Topic proportions θ of the prostate cancer dataset. We depict each of the classes with different colors. A) By clustering the genes BaLDA is able to use different topics to describe the 3 macroclasses; for example for the genes of the fourth cluster (K=4), the first topic describes the non-tumoral samples, the second topic the tumoral samples and the third the metastatic tumors. Again other clusters seem to highlight one of the three classes (the third cluster – K=3 – highlights metastatic using topic 2, etc). B) Comparison with [19] using a model with the same complexity. C) Comparison with [19] using the same number of topics. D) Comparison with [19] using the optimal topic number.

BaLDA has used to derive a highly informative and discriminant representation for microarray experiments. An experimental evaluation of the proposed methodologies on standard datasets confirms the effectiveness of the proposed techniques, also in comparison with other classification methodologies. Future works will focus on the biological interpretation of the results; it is evident that the interpretable topic representation of the expression matrix can be exploited to highlight genes strictly involved in the biological problem of interest, e.g. cancer or tumoral processes [1].

Acknowledgements

We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250).

References

1. Bicego, M., Lovato, P., Ferrarini, A., Delledonne, M.: Biclustering of expression microarray data with topic models. In: Proc. Int. Conf. on Pattern Recognition (2010)
2. Bicego, M., Lovato, P., Oliboni, B., Perina, A.: Expression microarray classification using topic models. In: ACM SAC - Bioinformatics and Computational Biology track (2010)
3. Blei, D., Ng, A., Jordan, M.: Latent dirichlet allocation. J. of Machine Learning Research 3, 993–1022 (2003)

4. Bosch, A., Zisserman, A., Munoz, X.: Scene classification via PLSA. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 517–530. Springer, Heidelberg (2006)
5. Brändle, N., Bischof, H., Lapp, H.: Robust DNA microarray image analysis. Machine Vision and Applications 15, 11–28 (2003)
6. Castellani, U., Perina, A., Murino, V., Bellani, M., Brambilla, P.: Brain morphometry by probabilistic latent semantic analysis. In: MICCAI (2010)
7. Chang, J., Boyd-Graber, J., Gerrish, S., Wang, C., Blei, D.: Reading the tea leaves: how humans interpret topic models. In: NIPS (2009)
8. Diaz-Uriarte, R., Alvarez de Andres, S.: Gene selection and classification of microarray data using random forest. BMC Bioinformatics 7(1), 3 (2006)
9. Frey, B., Dueck, D.: Clustering by passing messages between data points. Science 315, 972–976 (2007)
10. Hofmann, T.: Unsupervised learning by probabilistic latent semantic analysis. Mach. Learn. 42(1-2), 177–196 (2001)
11. Jaakkola, T., Haussler, D.: Exploiting generative models in discriminative classifiers. In: NIPS, pp. 487–493 (1999)
12. Jordan, M., Ghahramani, Z., Jaakkola, T., Saul, L.: An introduction to variational methods for graphical models. Machine Learning 37(2), 183–233 (1999)
13. Lee, J., Lee, J., Park, M., Song, S.: An extensive comparison of recent classification tools applied to microarray data. Computational Statistics & Data Analysis 48(4), 869–885 (2005)
14. Martins, A., Smith, N., Xing, E., Aguiar, P., Figueiredo, M.: Nonextensive information theoretic kernels on measures. J. of Machine Learning Research 10, 935–975 (2009)
15. Masada, T., Hamada, T., Shibata, Y., Oguri, K.: Bayesian multi-topic microarray analysis with hyperparameter reestimation. In: Proc. Int. Conf. on Advanced Data Mining and Applications (2009)
16. McLachlan, G., Bean, R., Peel, D.: A mixture model-based approach to the clustering of microarray expression data. BMC Bioinformatics 18(3), 413–422 (2002)
17. Osareh, A., Shadgar, B.: Classification and diagnostic prediction of cancers using gene microarray data analysis. J. of Applied Sciences 9(3) (2009)
18. Pomeroy, S., Tamayo, P., et al.: Prediction of central nervous system embryonal tumour outcome based on gene expression. Nature 415(6870), 436–442 (2002)
19. Rogers, S., Girolami, M., Campbell, C., Breitling, R.: The latent process decomposition of cdna microarray data sets. IEEE/ACM Trans. on Comp. Biology and Bioinformatics 2(2), 143–156 (2005)
20. Dhanasekaran, S., Barrette, T., et al.: Delineation of prognostic biomarkers in prostate cancer. Nature 23 412(6849), 822–826 (2001)
21. Statnikov, A., Aliferis, C., Tsamardinos, I., Hardin, D., Levy, S.: A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics 21(5), 631–643 (2005)
22. Valafar, F.: Pattern recognition techniques in microarray data analysis: A survey. Annals of the New York Academy of Sciences 980, 41–64 (2002)
23. Ying, Y., Li, P., Campbell, C.: A marginalized variational bayesian approach to the analysis of array data. BMC Proceedings 2(suppl. 4), S7 (2008)

Measuring the Quality of Shifting and Scaling Patterns in Biclusters

Beatriz Pontes[1], Raúl Giráldez[2], and Jesús S. Aguilar-Ruiz[2]

[1] Department of Computer Science, University of Seville
Avenida Reina Mercedes s/n, 41012, Sevilla, Spain
bepontes@us.es
http://www.lsi.us.es/~bepontes/index-en.html
[2] School of Engineering, Pablo de Olavide University
Ctra. de Utrera, km.1, 41013, Sevilla, Spain
{giraldez,aguilar}@upo.es
http://www.upo.es/eps/{giraldez,aguilar}

Abstract. The most widespread biclustering algorithms use the Mean Squared Residue (MSR) as measure for assessing the quality of biclusters. MSR can identify correctly shifting patterns, but fails at discovering biclusters presenting scaling patterns. Virtual Error (VE) is a measure which improves the performance of MSR in this sense, since it is effective at recognizing biclusters containing shifting patters or scaling patterns as quality biclusters. However, VE presents some drawbacks when the biclusters present both kind of patterns simultaneously. In this paper, we propose a improvement of VE that can be integrated in any heuristic to discover biclusters with shifting and scaling patterns simultaneously.

1 Introduction

The use of microarray techniques allows to study the activity of thousands of genes at a time, producing in this way a huge amount of data. Usually, the resulting data is organized in a matrix, called an expression matrix, where columns may represent genes and rows represent experimental conditions. An element of such expression matrix stands for the expression level of a given gene under a specific condition [3,18].

The interest in discovering knowledge from gene expression data has experimented an enormous increase with the development of microarray techniques. Biclustering [12] is becoming a popular data mining technique due to its ability to explore at the same time both dimensions of data, as opposed to clustering techniques [19], that can only use one dimension. In this sense, microarray is a suitable context for the application of biclustering techniques, since they can consider both genes and experimental conditions at extracting useful knowledge. Thus, in this context, a bicluster is a subset of genes under a subset of conditions. In particular, those biclusters where the subset of genes shows a common

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 242–252, 2010.

tendency under the subset of conditions are of special interest. In general, biclustering is much more complex than clustering [14]. In fact, finding significant biclusters in microarray data has been proven to be a NP-hard problem [17].

Cheng and Church [7] were the first in applying biclustering to gene expression data. They introduced one of the most popular biclustering algorithms that combines a greedy search heuristic for finding biclusters with a measure for assessing the quality of such biclusters. This measure, named *Mean Squared Residue* (MSR), has been used by many researchers who have proposed different heuristics for biclustering biological data. Aguilar et al. [2] developed an approach based on local nearness. Yang et al. [21] proposed an iterative algorithm for finding a predefined number of biclusters. Cano et al. [6] based their proposal on fuzzy technology and spectral clustering. Other approaches, such as Divina and Aguilar [10] and Bleuler et al. [4], have been based on evolutionary computation, while Bryan et al. [5] applied simulated annealing as heuristic. Recently, MSR has also been incorporated as cost function in multiobjective heuristics based on Particle Swarm Optimization [13] and Artificial Immune Systems [9].

Although MSR has been used in many proposals for finding biclusters, it nevertheless has been proven to be inefficient for finding certain types of biclusters in microarray data, especially when they present strong scaling tendencies [1]. Thus, we introduced in previous works an alternative measure named *Virtual Error* (VE) [15]. This measure is based on the concept of behavioural patterns, which aim at identifying common patterns between genes or conditions. VE is effective at recognizing biclusters containing shifting patters or scaling patterns as quality biclusters. However, it presents some drawbacks when both kind of patterns are presented simultaneously in the same bicluster. In this paper, we propose a novel variant of VE, called *Transposed Virtual Error* (VE^t), that allows to find biclusters that MSR and VE do not recognize as interesting ones.

This paper is organized as follows. In the next section, an description of the shifting and scaling patterns is given. We then provide a formal definition of VE^t in Section 3, followed by a formal analysis in Section 4, demonstrating its strength with regard to the behavioural patterns. In Section 5 we discuss the consequences of the theorems presented in this work, providing a test of the effect of the noise on the VE^t. Finally, we summarize the main conclusions in Section 6.

2 Behavioural Patterns in Gene Expression Data

When all the genes of a bicluster follow a similar tendency under the set of conditions, then such a bicluster may be potentially biologically interesting. Therefore, it seems to be a good idea to develop a quality measure for biclusters based on the idea of behavioural patterns for gene expression. Aguilar [1] presented an in-depth discussion on the possible patterns in gene expression data. He described formally two kind of patterns: shifting and scaling patterns. They have been defined using numerical relations among the values in a bicluster. Several works based their principle in the pattern concept in order to mine the data. Xu et al [20] propose a biclustering algorithm for mining shifting and scaling co-regulation patterns on gene expression data. Nevertheless, they do not provide a quality measure, but use a model-based heuristic instead. Furthermore, they are only able to identify global shifting and scaling patterns, while local ones seem to be more interesting since they depict the general situation [1]

Let $\mathcal{B}$ be a bicluster made up of I experimental conditions and J genes. Each element in $\mathcal{B}$ is represented by $b_{ij} \in \mathcal{B}$. This way, the bicluster $\mathcal{B}$ follows a *perfect shifting pattern* if its values can be obtained by adding a constant-condition number β_i to a typical value for each gene (π_j). β_i is said to be the *shifting coefficient* for condition i. In this case, the expression values in the bicluster fulfil the following equation:

$$b_{ij} = \pi_j + \beta_i \tag{1}$$

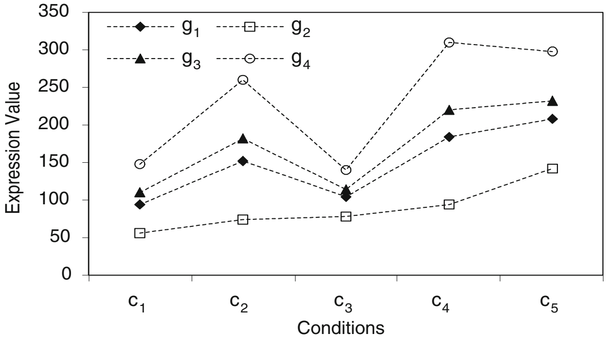
Similarly, a bicluster follows a *perfect scaling pattern* changing the additive value in the former equation by a multiplicative one. This new term α_i is called the *scaling coefficient*, and represents a constant value for each condition. The following equation defines whether a bicluster follows a perfect scaling pattern or not:

$$b_{ij} = \pi_j \times \alpha_i \tag{2}$$

Shifting and scaling patterns may be put together in a new kind of pattern called *combined pattern*. In fact, it is the most probable situation when working with real genetic data. In this situation, the expression values can be obtained using both coefficients, shifting and scaling coefficients. The equation that must be fulfilled by the values in this case can be represented by merging 1 and 2:

$$b_{ij} = \pi_j \times \alpha_i + \beta_i \tag{3}$$

Figure 1 shows an example of a bicluster obtaining from synthetic data. This is a typical visualization of bicluster, where conditions are represented in the x-axis, the values of gene expression are represented in the y-axis and each line is a gene. As we can see, there are four genes g_j (with $1 \leq j \leq 4$) and five conditions c_i (with $1 \leq i \leq 5$). This bicluster contains both shifting and scaling patterns. The matrices below describe the factor decomposition of the numerical values. Having a look at figure 1 we could say that genes g_1, g_3 and g_4 present a similar behaviour across the conditions, although g_4 has a different tendency between the last two conditions. On the contrary, the tendency of gene g_2 varies from the other genes, since its behaviour is always increasing across all the conditions. Gene g_2 is difficult to be associated to a perfect pattern visually because its shifting coefficients β_i are closed to the product $\pi_j \times \alpha_i$. Therefore, identifying biclusters with both shifting and scaling patterns might be a difficult task due to the complexity inherent in equation 3.



$$\mathcal{B} = \begin{pmatrix} 95 & 56 & 110 & 149 \\ 152 & 74 & 182 & 260 \\ 104 & 78 & 114 & 140 \\ 185 & 94 & 220 & 311 \\ 208 & 143 & 233 & 298 \end{pmatrix} = \begin{pmatrix} 25 \times 3 + 20 & 12 \times 3 + 20 & 30 \times 3 + 20 & 43 \times 3 + 20 \\ 25 \times 6 + 2 & 12 \times 6 + 2 & 30 \times 6 + 2 & 43 \times 6 + 2 \\ 25 \times 2 + 54 & 12 \times 2 + 54 & 30 \times 2 + 54 & 43 \times 2 + 54 \\ 25 \times 7 + 10 & 12 \times 7 + 10 & 30 \times 7 + 10 & 43 \times 7 + 10 \\ 25 \times 5 + 83 & 12 \times 5 + 83 & 30 \times 5 + 83 & 43 \times 5 + 83 \end{pmatrix}$$

$$\begin{array}{rl} & \begin{array}{cccc} \pi_1 & \pi_2 & \pi_3 & \pi_4 \end{array} \\ \{\pi_j\} = & \{\begin{array}{cccc} 25 & 12 & 30 & 43 \end{array}\} \\ & \begin{array}{ccccc} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \end{array} \\ \{\alpha_i\} = & \{\begin{array}{ccccc} 3 & 6 & 2 & 7 & 5 \end{array}\} \\ & \begin{array}{ccccc} \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 \end{array} \\ \{\beta_i\} = & \{\begin{array}{ccccc} 20 & 2 & 54 & 10 & 83 \end{array}\} \end{array}$$

Fig. 1. Bicluster containing perfect shifting and scaling patterns

3 Transposed Virtual Error

Virtual Error (VE) had been designed as an evaluation measure for biclusters [15]. VE is based on the concepts of shifting and scaling patterns and is capable of identifying both kind of patterns within biclusters, although not simultaneously. This way, VE improves MSR effectiveness, since the last one can only recognize shifting patterns. The basic idea behind VE is to measure the extent to which the genes in a bicluster are similar to the general tendency. The general tendency is represented by a *Virtual Gene* which is created taking into account the values for every gene across the experimental conditions, but trying to capture the general behaviour with independence of the concrete numerical values. VE will have a lower value for those biclusters in which its genes are closer to the virtual gene. This is due to the fact that VE computes the numerical differences between each standardized gene and the standardized virtual gene. Therefore, the better a bicluster is, the lower its VE value will be. Furthermore, it is obvious that VE will always be greater or equal than zero.

VE has been used in various evolutionary algorithms in order to find one hundred biclusters in several gene expression matrices [15,11]. These two previous works have allowed us finding interesting biclusters that could not have been obtained using MSR alone. Furthermore, the VE value for biclusters with perfect shifting and scaling patterns seems to be very close to zero [16] (magnitude of 10^{-15}). Nevertheless, VE cannot be proven to recognize both kind of patterns simultaneously.

In this work, we present an enhanced version of VE, named VE^t, from *Transposed Virtual Error*. We analytically prove that VE^t is zero for those biclusters with perfect shifting and scaling patterns. This variation of VE has been motivated by [8], where several numeric transformations have been applied to the data in order to detect both kind of patterns.

VE^t is computed similarly to VE but considering the transposed bicluster. The idea here would be to create a *Virtual Condition*, instead of a virtual gene, and measure the differences between the standardized values for every condition and the standardized virtual condition. In the following, we explain how to create the virtual condition for a certain bicluster $\mathcal{B}$, in order to compute VE^t afterwards.

Definition 1 (Virtual Condition). *Given a bicluster $\mathcal{B}$ with I conditions and J genes, we define its virtual condition as a collection of J elements ρ_j, each of them defined as the mean of the j^{th} column: $\rho_j = \frac{\sum_{i \in I} b_{ij}}{I}$, where $b_{ij} \in \mathcal{B}, 1 \leq i \leq I$ and $1 \leq j \leq J$.*

This way, each element of the virtual condition represents a meaningful value for all the conditions, regarding each gene. Once the virtual condition has been created, the next task would consist of quantifying the way in which all the experimental conditions in the bicluster are similar to it. In order to perform an appropriate comparison, we first carry out a standardization of the virtual condition and of every experimental condition in the bicluster. This standardization allows us to capture the differences among the tendencies, with independence of the numerical values.

Definition 2 (Standardization). *We define the standardized bicluster $\hat{\mathcal{B}}$ from bicluster $\mathcal{B}$ as a new bicluster in which its elements $\hat{b_{ij}}$ are defined by $\hat{b_{ij}} = \frac{b_{ij} - \mu_{c_i}}{\sigma_{c_i}}$, where σ_{c_i} and μ_{c_i} represent the standard deviation and the arithmetic average of all the expression values for condition i, respectively.*

It has already been said that the virtual condition needs also to be standardized. Equation 4 shows how the values of the standardized virtual condition are obtained, where ρ_j refers to the virtual condition value for gene j, while μ_ρ and σ_ρ refer to the average and the deviation of the values of the virtual condition, respectively.

$$\hat{\rho_j} = \frac{\rho_j - \mu_\rho}{\sigma_\rho} \tag{4}$$

Definition 3 (Transposed Virtual Error). *Given a bicluster $\mathcal{B}$, and the virtual condition ρ, Transposed Virtual Error (VE^t) can be defined as the mean of the numerical differences between each standardized condition and the values of the standardized virtual condition for each gene:*

$$VE^t(\mathcal{B}) = \frac{1}{I \cdot J} \sum_{i=1}^{i=I} \sum_{j=1}^{j=J} (\hat{b_{ij}} - \hat{\rho_j}) \tag{5}$$

Next, we present three theorems and their proofs that demonstrate the strength of VE^t with regard to the shifting and scaling patterns.

4 Analysis

This section includes formal proofs that bear out the hypothesis that VE^t is zero for those biclusters with perfect shifting and scaling patterns, either separately or simultaneously.

Theorem 1. *A bicluster presenting a perfect shifting pattern has VE^t equal to zero.*

Proof. Let $\mathcal{B}$ be a bicluster with a perfect shifting pattern, then it is possible to refer to its elements as $b_{ij} = \pi_j + \beta_i$. Applying two simple arithmetic properties[1], the mean and the deviation for each condition c_i can be expressed by:

$$\mu_{c_i} = \mu_\pi + \beta_i \quad ; \quad \sigma_{c_i} = \sigma_\pi$$

where μ_π and σ_π represent the mean and the deviation of the π values, respectively. Using these results we obtain the standardizes values for b_{ij}:

$$\hat{b}_{ij} = \frac{b_{ij} - \mu_{c_i}}{\sigma_{c_i}} = \frac{\pi_j + \beta_i - \mu_\pi - \beta_i}{\sigma_\pi} = \frac{\pi_j - \mu_\pi}{\sigma_\pi}$$

Combining the former properties[1] it is easy to express the mean and standard deviation for the virtual condition as:

$$\mu_\rho = \mu_\pi + \mu_\beta \quad ; \quad \sigma_\rho = \sigma_\pi$$

Finally, the standardized values for the virtual condition are the following:

$$\hat{\rho}_j = \frac{\rho_j - \mu_\rho}{\sigma_\rho} = \frac{\pi_j + \mu_\beta - \mu_\pi - \mu_\beta}{\sigma_\pi} = \frac{\pi_j - \mu_\pi}{\sigma_\pi} = \hat{b}_{ij}$$

As it can be seen above, the standardized virtual condition is equal to all the real conditions after being standardized. Therefore, VE^t has been proven to be zero for those biclusters with perfect shifting patterns. ∎

Theorem 2. *A bicluster presenting a perfect scaling pattern has VE^t equal to zero.*

Proof. Let $\mathcal{B}$ be a bicluster following a perfect scaling pattern, then its elements can be expressed by $b_{ij} = \pi_j \times \alpha_i$. Following the same reasoning that in the former proof, the mean and deviation of each condition c_i are:

$$\mu_{c_i} = \alpha_i \times \mu_\pi \quad ; \quad \sigma_{c_i} = \alpha_i \times \sigma_\pi$$

From these results we obtain the standardized values for b_{ij}:

$$\hat{b}_{ij} = \frac{b_{ij} - \mu_{c_i}}{\sigma_{c_i}} = \frac{\pi_j \times \alpha_i - \alpha_i \times \mu_\pi}{\alpha_i \times \sigma_\pi} = \frac{\pi_j - \mu_\pi}{\sigma_\pi}$$

Next we obtain the mean and deviation for the values of the virtual condition:

$$\mu_\rho = \mu_\pi \times \mu_\alpha \quad ; \quad \sigma_\rho = \mu_\alpha \times \sigma_\pi$$

[1] Being $f(x) = g(x) \times c_1 + c_2$, the properties related to the arithmetic mean ($\mu_{f(x)}$) and the standard deviation ($\sigma_{f(x)}$) of $f(x)$ are the following: $\mu_{f(x)} = \mu_{g(x)} \times c_1 + c_2$ and $\sigma_{f(x)} = \sigma_{g(x)} \times c_1$.

And finally the standardized values for the virtual condition are:

$$\hat{\rho}_j = \frac{\rho_j - \mu_\rho}{\sigma_\rho} = \frac{\pi_j \times \mu_\alpha - \mu_\pi \times \mu_\alpha}{\mu_\alpha \times \sigma_\pi} = \frac{\pi_j - \mu_\pi}{\sigma_\pi} = \hat{b}_{ij}$$

As in the previous proof, we obtain that the standardized values for the virtual condition are equal to de standardized values for all the real experimental conditions. As a consequence, VE^t will be zero for every bicluster with a perfect scaling pattern. ■

Theorem 3. *A bicluster presenting a perfect combined pattern (shifting and scaling) has VE^t equal to zero.*

Proof. If $\mathcal{B}$ contains a perfect combined pattern, its values can be represented by $b_{ij} = \pi_j \times \alpha_i + \beta_i$. Using the same arithmetic properties as in the former proves, the mean and deviation for each condition c_i are:

$$\mu_{c_i} = \alpha_i \times \mu_\pi + \beta_i \quad ; \quad \sigma_{c_i} = \alpha_i \times \sigma_\pi$$

And the standardized values for b_{ij} can be expressed as:

$$\hat{b}_{ij} = \frac{b_{ij} - \mu_{c_i}}{\sigma_{c_i}} = \frac{\pi_j \times \alpha_i + \beta_i - \alpha_i \times \mu_\pi + \beta_i}{\alpha_i \times \sigma_\pi} = \frac{\pi_j - \mu_\pi}{\sigma_\pi}$$

The mean and deviation for the virtual condition are the following:

$$\mu_\rho = \mu_\pi \times \mu_\alpha + \mu_\beta \quad ; \quad \sigma_\rho = \mu_\alpha \times \sigma_\pi$$

And the standardized values for the virtual condition:

$$\hat{\rho}_j = \frac{\rho_j - \mu_\rho}{\sigma_\rho} = \frac{\pi_j \times \mu_\alpha + \mu_\beta - \mu_\pi \times \mu_\alpha - \mu_\beta}{\mu_\alpha \times \sigma_\pi} = \frac{\pi_j - \mu_\pi}{\sigma_\pi} = \hat{b}_{ij}$$

Again, the standardized values for the virtual condition match up with the standardized values for the original conditions. Therefore, VE^t will also be zero for those biclusters following a perfect shifting and scaling pattern. ■

These results confirm that VE^t is the first measure up to the date capable of recognizing combined patterns in gene expression data. While MSR is only capable of detecting shifting patterns, and VE cannot recognize both kind of patterns simultaneously, VE^t has been proven to go beyond the other two measures.

5 Discussion

In this section, we discuss the use of VE^t for bicluster evaluation. In particular, we study the value of VE^t for those biclusters in which the presence of patterns is not perfect. That is, when the tendency of the data in a bicluster is similar to a perfect pattern but does not completely match with the equation 3.

In order to check the behaviour of VE^t whenever a bicluster does not follow a perfect pattern, we add an additive term ε_{ij} to the combined pattern equation.

The meaning of this new term corresponds to the error made by the assumption that the bicluster can be represented by a perfect pattern.

$$b_{ij} = \pi_j \times \alpha_i + \beta_i + \varepsilon_{ij} \tag{6}$$

It is possible therefore to study the variations produced to VE^t depending on the values of ε_{ij}. Nevertheless, it is not so simple due to the huge amount of situations depending on the distribution and the magnitude of the ε_{ij} values in the data matrix.

In two specific situations the value of VE^t will not be affected when the errors could be included in the former equation 6. These two cases correspond to those in which ε_{ij} values are either a constant or constants per conditions (rows). In both cases it is possible to eliminate the term ε_{ij} from the equation, since it can be considered to be a part of β_i.

Nevertheless, the cases in which ε_{ij} cannot not be included in the perfect pattern equation are very difficult to study analytically. For this reason, we have performed a test to check the tendency of the VE^t values with regard to the error values. This test consist of the addition of random errors to a synthetic bicluster with perfect shifting and scaling patterns. The original bicluster is the one shown in Fig. 1. Specifically, we have generated 100 synthetic biclusters adding random errors to the bicluster in the figure, and we have repeated this process 200 different times, varying the amplitude of the errors from one time to another. We start adding negative errors in the range of $[-10, 0]$, and obtain 100 different biclusters. Then we decrease the amplitude by 0.1 and repeat the process (range $[-9.90, 0]$). Once the amplitude of the errors has reached the zero value, we start again generating biclusters with positive errors, increasing the amplitude from 0.1 up to 10. The whole process produced 100 sets of 100 biclusters with negative errors and 100 sets of 100 biclusters with positive errors (built using the same strategy as for negative). Therefore, the random errors have been drawn from an uniform distribution corresponding to the ranges. Note that the type of the error values is a double type. This introduces more diversity in the distribution of the error data.

Within the process, we evaluate each produced bicluster using the three measures: MSR, VE and VE^t. Then we obtain the mean of each measure for each group of 100 biclusters of the same range of errors. This data has been represented in Figs. 2, 3 and 4, where the x-axis represents the mean of the error for each amplitude (this value matches up with the value in the middle of the range of errors) and the y-axis corresponds to the mean of the specific measure for each figure.

From Fig. 2 it is possible to observe that VE^t presents a linear decreasing tendency in relation to the amount of error in a bicluster. In other words, the similar a bicluster is to a perfect pattern, the lower its VE^t value will be, and

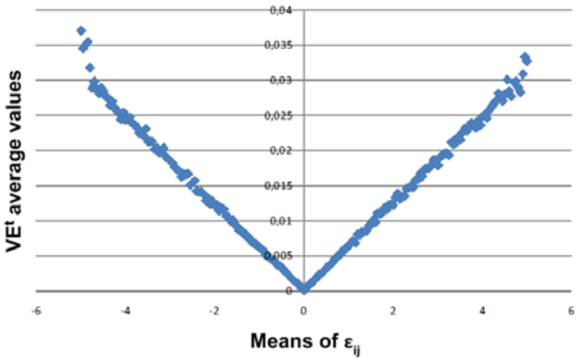


Fig. 2. VE^t behaviour in biclusters with errors

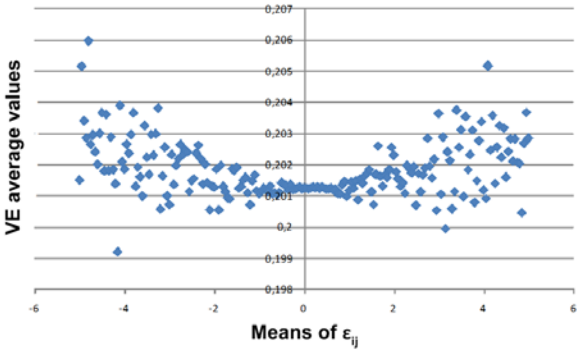


Fig. 3. VE behaviour in biclusters with errors

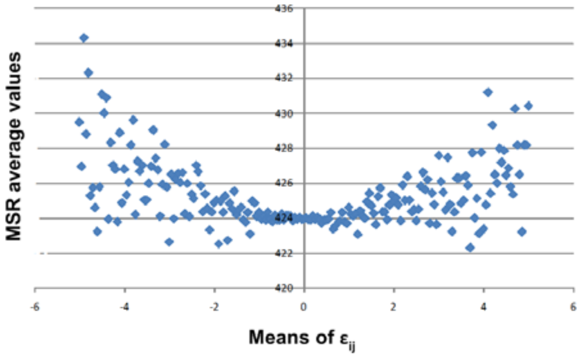


Fig. 4. MSR behaviour in biclusters with errors

we can establish a linear relationship between VE^t and the amount of error. Nevertheless, it is not possible to come to the same conclusion for either VE or MSR. Figs. 3 and 4 depict the connection of the errors with VE and MSR, respectively. Although the general tendency seems to be that both measures are higher for biclusters with higher error values, we cannot establish any correspondence between them. In both figures it is possible to see some cases in which the mean of the biclusters with errors is lower than the original bicluster.

As a conclusion, VE^t outperforms both MSR and VE efficacy for identifying behavioural patterns in synthetic data. Our expectations are that this behaviour would be extensive to real gene expression data.

6 Conclusions

This work introduces an enhanced version of a previous measure for evaluating biclusters from gene expression data. This new variant, named VE^t, allow finding biclusters with both shifting and scaling patterns simultaneously in gene expression data. No previous evaluation measure for biclusters is able of identifying this kind of pattern, for this reason we are sure VE^t constitutes an important contribution to the topic.

This paper also includes analytical proofs which demonstrate the capability of VE^t for detecting any kind of perfect pattern in gene expression data. Furthermore, we have also proved that VE^t presents a linear relationship with the amount of error in a bicluster.

For future work, we have planned to use VE^t together with an evolutionary framework in order to search for biclusters in gene expression data. The obtained results will be compared to those obtained by similar heuristics and evaluation measures. Biological validation of the results will also be performed in order to validate our approach.

Acknowledgement

This research is supported by the Spanish Ministry of Science and Technology under grant TIN2007-68084-C02-00 and the Junta de Andalucía Research Program.

References

1. Aguilar-Ruiz, J.S.: Shifting and scaling patterns from gene expression data. Bioinformatics 21, 3840–3845 (2005)
2. Aguilar-Ruiz, J.S., Rodriguez, D.S., Simovici, D.A.: Biclustering of gene expression data based on local nearness. In: Proceedings of EGC 2006, Lille, France, pp. 681–692 (2006)
3. Baldi, P.: DNA Microarrays and Gene Expression: From Experiments to Data Analysis and Modeling. Cambridge University Press, Cambridge (2002)

4. Bleuler, S., Prelić, A., Zitzler, E.: An EA framework for biclustering of gene expression data. In: Congress on Evolutionary Computation (CEC-2004), pp. 166–173. IEEE, Los Alamitos (2004)
5. Bryan, K., Cunningham, P., Bolshakova, N.: Application of simulated annealing to the biclustering of gene expression data. IEEE Transactions on Information Technology on Biomedicine (2006)
6. Cano, C., Adarve, L., López, J., Blanco, A.: Possibilistic approach for biclustering microarray data. Computers in Biology and Medicine 37(10), 1426–1436 (2007)
7. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proceedings of the 8th International Conference on Intellingent Systemns for Molecular Biology, La Jolla, CA, pp. 93–103 (2000)
8. Cho, H., Dhillon, I.S.: Effect of data transformation on residue. Technical report (2007)
9. Coelho, G.P., de Franca, F.O., Zuben, F.J.V.: Multi-objective biclustering: When non-dominated solutions are not enough. Journal of Mathematical Modelling and Algorithms 8(2), 175–202 (2009)
10. Divina, F., Aguilar-Ruiz, J.S.: Biclustering of expression data with evolutionary computation. IEEE Transactions on Knowledge & Data Engineering 18(5), 590–602 (2006)
11. Divina, F., Aguilar-Ruiz, J.S., Pontes, B., Giráldez, R.: An effective measure for assessing the quality of biclusters (in Press, 2010)
12. Hartigan, J.: Direct clustering of a data matrix. Journal of the American Statistical Association 67(337), 123–129 (1972)
13. Liu, J., Li, Z., Hu, X., Chen, Y.: Biclustering of microarray data with mospo based on crowding distance. BMC bioinformatics 10(suppl. 4), S9+ (2009)
14. Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. IEEE Transactions on Computational Biology and Bioinformatics 1, 24–25 (2004)
15. Pontes, B., Divina, F., Giráldez, R., Aguilar-Ruiz, J.S.: Virtual error: A new measure for evolutionary biclustering. In: Fifth European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics (EvoBio 2007), pp. 217–222 (2007)
16. Pontes, B., Giráldez, R., Divina, F., Martínez-Álvarez, F.: Evaluación de biclusters en un entorno evolutivo. In: IV Taller nacional de minería de datos y aprendizaje (TAMIDA), pp. 1–10 (2007)
17. Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. Bioinformatics 18, 136–144 (2002)
18. Tilstone, C.: Dna microarrays: Vital statistics. Nature 424, 610–612 (2003)
19. Wang, H., Wang, W., Yang., J., Yu, P.S.: Clustering by pattern similarity in large data sets. In: ACM SIGMOD International Conference on Management of Data, Madison, WI, pp. 394–405 (2002)
20. Xu, X., Lu, Y., Tung, A.K.H., Wang, W.: Mining shifting-and-scaling co-regulation patterns on gene expression profiles. In: 22nd International Conference on Data Engineering (ICDE'06), pp. 89–99 (2006)
21. Yang, J., Wang, H., Wang, W., Yu, P.S.: An improved biclustering method for analyzing gene expression profiles. International Journal on Artificial Intelligence Tools 14, 771–790 (2005)

Frequent Episode Mining to Support Pattern Analysis in Developmental Biology

Ronnie Bathoorn, Monique Welten, Michael Richardson,
Arno Siebes, and Fons J. Verbeek

Imaging & BioInformatics, LIACS, Leiden University, The Netherlands (MW, MR, FJV)
Distributed Databases, Computer Science, Utrecht University, The Netherlands (RB, AS)
ronnie@cs.uu.nl, fverbeek@liacs.nl

Abstract. We introduce a new method for the analysis of heterochrony in developmental biology. Our method is based on methods used in data mining and intelligent data analysis and applied in, e.g., shopping basket analysis, alarm network analysis and click stream analysis. We have transferred, so called, frequent episode mining to operate in the analysis of developmental timing of different (model) species. This is accomplished by extracting small temporal patterns, i.e. episodes, and subsequently comparing the species based on extracted patterns. The method allows relating the development of different species based on different types of data. In examples we show that the method can reconstruct a phylogenetic tree based on gene-expression data as well as using strict morphological characters. The method can deal with incomplete and/or missing data. Moreover, the method is flexible and not restricted to one particular type of data: i.e., our method allows comparison of species and genes as well as morphological characters based on developmental patterns by simply transposing the dataset accordingly. We illustrate a range of applications.

Keywords: frequent episode mining, heterochrony, pattern analysis, developmental biology.

1 Introduction

The relation between evolution and development is intriguing [11,12] and considered essential for gaining understanding in the tree of life. Heterochrony, defined as the change of timing in events in development leading to changes in size and shape of species, facilitates analyzing differences in species. The key goal in heterochrony analysis is to relate evolutionary distance between species to changes in timing of developmental events. Tools to analyze developmental timing in a quantitative way have shown not to perform adequately for large datasets. In addition, for assessment, a relative timing is required and such is not present in existing computational approaches. Therefore, complementary to other methods, such as event-pairing [11] and Search-based Character Optimization [15], we developed a method for heterochrony analysis that includes efficient extraction of developmental patterns and at the same time allows using different types of data, e.g. morphological and gene-expression, in a universal manner. To that end we propose an analysis of

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 253–263, 2010.

developmental sequences based on, so called, episodes [13]. Episodes are small, partially ordered, sets of events that frequently occur in the data. A collection of episodes extracted from a developmental sequence provides a good basis for further analysis of that sequence. For our method to run efficiently a special data structure is required to accomplish fast updates on the extracted patterns. We, therefore, propose a data structure referred to as the *episode tree* which is specifically designed for and tailored to this kind of application.

Our analysis starts with a dataset containing developmental sequences (cf. § 2.2) and from this dataset an episode tree is created by sliding a window over the developmental sequences; all episodes found in this time window are added to the episode tree. Subsequently, a distance measure, based on the concept of *heterochrony*, is used between the entities in our dataset (species). After computation of the distances and clustering based on these distances, results are obtained and visualized as cladogram; typically showing evolutionary distance between species.

Experiments with artificial, morphological and gene expression datasets are used to illustrate the scope of this method. In each of the experiments the entities we compare to each other can be different, i.e. clades, species or genes. Importantly, using a gene expression dataset as input, results in a cladogram similar to those from biological literature. For our experiments we consider gene expression as extracted from patterns of gene expression from "*in situ*" hybridization, these are directly related to morphological characters. At this point, Micro Arrays gene expression patterns are not considered.

2 Materials and Methods

Here, we will describe the *Frequent Episodes mining in Developmental Analysis* (FEDA). The method is centered on a database (MySQL [2]) that contains the data to be analyzed as well as the patterns extracted. This approach facilitates the selection of interesting patterns for further analysis. The data were extracted from the literature and imported in the database (Fig. 1A). The software runs on a standard PC.

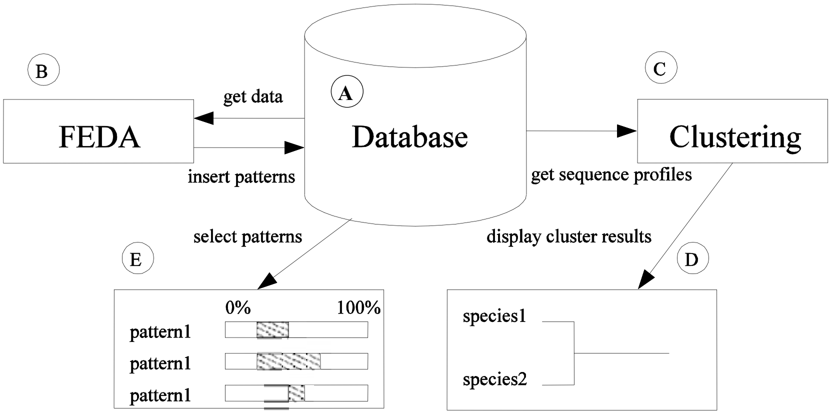


Fig. 1. Overview of FEDA architecture centered on a database. (A) Data import in the database. (B) FEDA finds frequent episodes and inserts these back in database. (C &D) Visual output, like clustering developmental profiles (C & D) or a pattern shift diagram (E) from frequent episodes.

2.1 Finding the Episodes

We propose a method for finding sequence heterochrony in developmental sequences (Fig. 1B). Using small frequently occurring patterns, called episodes, we try to find differences between developmental sequences. In order to have an unequivocal idea of the major concepts we define the core entities.

Definition 1 *Developmental Sequence*: A developmental sequence is an ordered list of pairs. A pair consists of a developmental events and a timepoint. The list is ordered on the timepoints and describes the timing of these events within one species.

Definition 2 *Episode*: An episode [13] is a small ordered set of events that is frequent over all developmental sequences.

Definition 3 *Frequency*: The frequency of an episode is the total sum of the occurrences (cf. Def. 4) of this specific episode in all sequences. For each occurrence its size is equal to or smaller than the maximum episode size.

Definition 4 *Occurrence*: For an episode to occur in a sequence, the events in this sequence need to be strictly ordered in the same order as the events in the episode. Events in between that are not part of the episode may exist. Consequently, gaps between events in an episode can exist; i.e., events in an occurrence do not have to be contiguous.

Definition 5 *Episode Size*: The size of an episode occurring in a sequence s is the size of the smallest subsequence s' of our sequence in which the episode can still be matched. Such "match" is called an *occurrence* of the episode in s. To limit the amount of episodes that can be identified, the size of the episodes has been restricted. The maximal episode size is the upper bound on the episode size.

FEDA uses the episode tree to store this collection of episodes together with their frequency.

Definition 6 *Episode Tree*: An episode tree is a prefix-tree data structure on the episodes with the following features:

1. consists of nodes and children
2. the tree has an empty root node; the start of all the episodes
3. each node has zero or more child nodes and each node contains: *an event, a frequency* and *a binary list*
4. a node with no child nodes is called a leaf

The FEDA algorithm starts with a given maximal episode size and an empty episode tree as parameters. FEDA processes all the sequences in the data and integrates all occurrences of the episodes identified in each sequence. This results in a collection of all episodes with the given maximal size together with their frequency. The root node is the empty episode from which all other episodes are extended. All children of this root-node are episode trees containing episodes that start with the event contained in this node. In addition, each of the children stores a binary list with length equal to the number of sequences in the dataset and it holds a 1(*true*) at position 1 if the episode is found in the first sequence. This is the same for the other sequences in the dataset. In an episode tree the events found in all nodes passed in the path from the root to another node is an episode. An example of an episode tree is depicted in Figure 2.

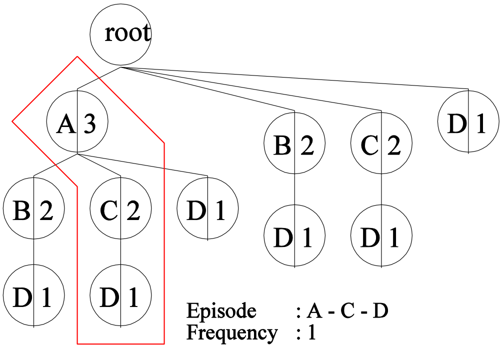


Fig. 2. An Episode Tree; the root is the start of all episodes contained in the tree. The highlighted path from the root to a leaf (end node) is the episode A-C-D with a frequency 1, as stored in the last node of the episode.

2.2 Episodes in Heterochrony Analysis

After computation of all the frequent episodes it is exactly known which patterns occurred in which developmental sequence and the frequency of each pattern in all the developmental sequences. From the data developmental profiles are constructed; these are defined as:

Definition 7 *Developmental Profile*: A developmental profile is a vector that exactly shows which episodes where found in a given developmental sequence. The number of elements in this vector is equals the number of episodes found by FEDA. All episodes are indexed for the developmental profile. The value at each index is *true* if the episode was found or *false* if not found.

The developmental profiles can be used in standard clustering algorithms while still being able to capture the temporal dependencies in the developmental sequence. Furthermore, filters can be used to control the size of the profiles. A possible filter is to use all maximal frequent episodes instead of all frequent episodes and thereby choose a minimal frequency as a threshold.

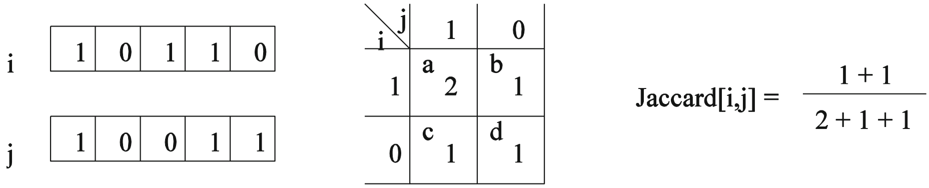
Definition 8 *Maximal Frequent Episodes*: An episode is maximal frequent if it is not part of a larger frequent episode, i.e. a collection of maximal frequent episodes does not contain small episodes that are part of other larger episodes in the collection.

2.3 Clustering of Developmental Profiles

After the episode mining step, a developmental profile is obtained, indicating which episodes have been found in each of the sequences. This profile is used as a feature vector describing each of the sequences. The similarity/dissimilarity between sequences in the data is visualized by application of clustering on the developmental profiles (Fig. 1C). The measurement of the distance between sequences requires a specific distance measure that excludes, in the distance, those episodes not present in both of the developmental profiles. The choice of the distance measure is motivated by the fact that an episode not being present in both developmental profiles is not contributing information on the biological difference between these two developmental profiles. This feature is typically expressed in the *Jaccard* distance [9], defined as:

$$Jaccard[i, j] = \frac{b + c}{a + b + c},$$

where both *i* and *j* are developmental profiles; *a* is the number of episodes present in both *i* and *j*, *b* is the number of episodes present only in *i*, *c* those only in *j*. In addition, *d* represents the episodes in none of the two profiles, *d* is not used in the computation but completes a cross table in the analysis of the profiles (Fig. 3). It is easily seen that the *Jaccard* distance is a normalized figure; 0 for b,c=0 and 1 for a=0.



i	1	0	1	1	0
j	1	0	0	1	1

i \ j	1	0
1	a 2	b 1
0	c 1	d 1

$$\text{Jaccard[i,j]} = \frac{1+1}{2+1+1}$$

Fig. 3. Shown from left to right: the profiles *i* and *j*; a cross table recording the number of episodes shared by i and j (a) all episodes possessed by only one profile (b and c) and those contained in none of the profiles (d); the computation of the *Jaccard* distance.

In our analysis, the *Jaccard* distance reflects a relevance of the identified episodes. The *Jaccard* distance is used constructing a dissimilarity matrix by computing it for all possible pairs of two species (Fig. 3). Subsequently, this matrix is used in the clustering. The agglomerative hierarchical clustering with complete linkage [10] is used; this is an unsupervised clustering method which initiates with a cluster for each of the entities present [1, *hclust*]. Subsequently, the two clusters that are closest are merged in a larger cluster and merging continues until all entities are in one cluster. The distance between two merged clusters is computed using complete linkage; i.e., all distances between all pairs of entities are computed and the largest of these distances is considered the distance between the two clusters. From the clustering result a cladogram can be derived (Fig. 1D) visualizing the distances between all sequences in the dataset. The root of this cladogram is the point at which all species are joined in one large cluster whereas the leaves represent clusters containing only one species.

3 Results

As a proof of principle to show the different aspects of the method we present results of a number of experiments using three datasets: a small artificial dataset to demonstrate the method (§ 3.1), a dataset of morphological events over time (§ 3.2) and a gene expression dataset (§ 3.4). The data are obtained through literature analysis. All the experiments produce a taxonomy tree that is compared to literature.

3.1 Artificial Data

We will illustrate our method with a simple dataset, obtained from the literature [15], consisting of 3 taxa and one outgroup. It illustrates that FEDA treats the episodes as dependent features and as such is not prone to errors found in event pairing [11,12] where events are treated independently and as a result shifts in timing cannot be attributed to one event. Feature dependency is preserved in the ordering of events in the episodes (Table 1). We start with building a list of all episodes found in this dataset; we adhere to only adding episodes that are found in the data instead of all possible combinations of events that are found in our dataset. Table 2 contains a list of all the episodes that were found in each sequence resulting in a developmental profile for all 4 sequences. In Table 2 a “1” indicates that the episode was present in the sequence and a “0” indicates it was absent. Next, the dissimilarity matrix between all sequences is computed by summing all differences between each pair of profiles. For taxa 1 and 2 this results in 6 differences in their profiles (AB, AC, BA, CA, ABC, BCA). Repeating this for all sequences in the example results in a distance matrix (Table 1B). This distance matrix shows that the distances between Taxa 1, 2 and 3 are all 0.86; the distance is computed using the *Jaccard* distance for all pairs of taxa. All pairs have 6 episodes for which the occurrence is different and 1 episode that is present in both taxa, resulting in a dissimilarity score of 6/7 between all the taxa, and a dissimilarity of 0 between the Outgroup and Taxon 1.

Finally, agglomerative clustering with complete linkage is applied, using the previously obtained distance matrix. The result is presented in Figure 4. The cladogram is realized by starting with all taxa in different clusters at the bottom of the cladogram and then merging clusters of the closest taxa. At completion, we end up with all taxa and the outgroup in one cluster at the top of the cladogram.

Table 1. (**A**) Dataset of 3 taxons and 1 outgroup (**B**) Distance Matrix showing the distance between each pair of taxa based on Jaccard distance. All taxa are equally close to one another.

A	Sequence
Outgroup	ABC
Taxon 1	ABC
Taxon 2	BCA
Taxon 3	CAB

B	Out	T 1	T 2	T
Out	0			
T 1	0	0		
T 2	0.86	0.86	0	
T 3	0.86	0.86	0.86	0

Table 2. Developmental Profiles recording the frequent episodes that occur in each taxon as well as the total number of times each episode occurs in the dataset.

	A	A	B	B	C	C	A	B	C
Outgroup	1	1	0	1	0	0	1	0	0
Taxon 1	1	1	0	1	0	0	1	0	0
Taxon 2	0	0	1	1	1	0	0	1	0
Taxon 3	1	0	0	0	1	1	0	0	1
Totals	3	2	1	3	2	1	2	1	1

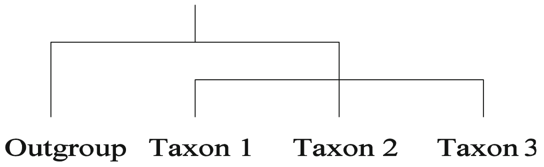


Fig. 4. The cladogram resulting from clustering the taxa based on the in Table 1B

3.2 Sequences of Morphological Characters in Development

Next, we illustrate FEDA with a more complex dataset [11] containing timed sequences of morphological events. Each event is taken from the development of one species. An event happens only once in a species. The dataset contains 14 entities (species), and one developmental sequence containing morphological events per entity (Fig. 5).

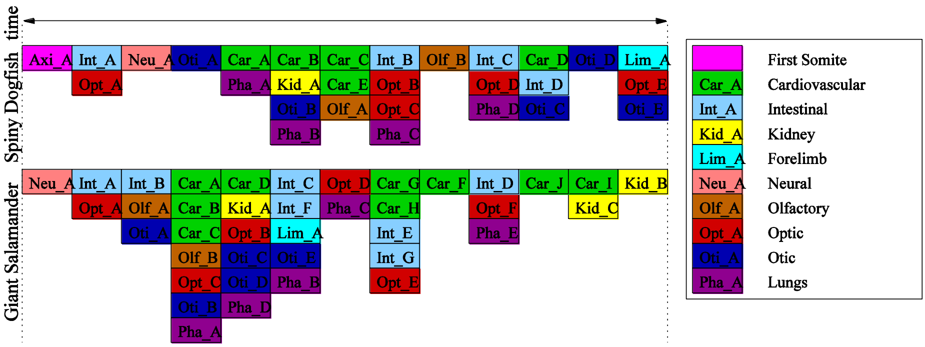


Fig. 5. Part of a recording of 2 developmental sequences presenting morphogical events over time. Here only spiny dogfish and giant salamander are depicted (dataset contains 14 species).

If all frequent episodes were used in the clustering this would result in long runtimes and therefore the frequent episode set is reduced. Using only maximal frequent episodes (cf. Def. 8) in our experiments reduces the number of episodes in the clustering, as only the larger episodes are extracted. The window size was increased to obtain a sufficient number of features to cluster the data. For this particular dataset the parameters for FEDA were set to a window size of 8 and a frequency threshold of 0.05, resulting in obtaining 983 episodes. In Figure 6 the resulting cladogram is depicted. The clustering is almost the same as the Taxonomy common tree [3], only minor differences are seen in the amphibians. The results obtained from event-pairing on this dataset [11] show the same pattern acknowledging that the granularity of the dataset is, actually, insufficient.

3.3 Relative Timescale in Development

To allow linking patterns between different species, a relative timescale is introduced and used in the computations. This timescale is based on percentage of development

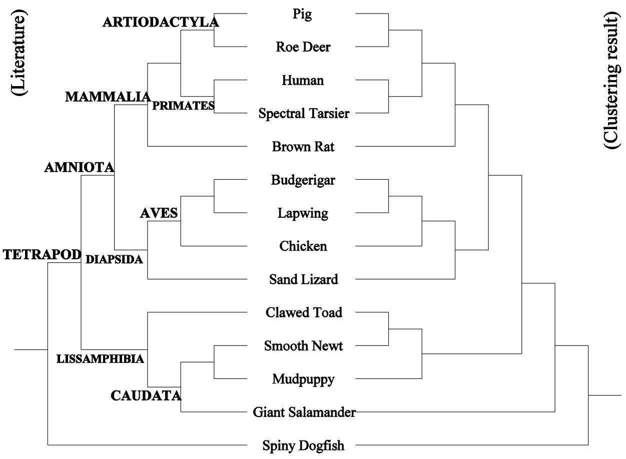


Fig. 6. Cladogram of the results computed by FEDA from a dataset of morphological events (right) compared to the taxonomy common tree from the NCBI (left)

of the species under study [23] and events are linked relative to the developmental scale this species. E.g. gene *tbx5* is active in Zebrafish in [5% - 10%] of development (Fig. 7).

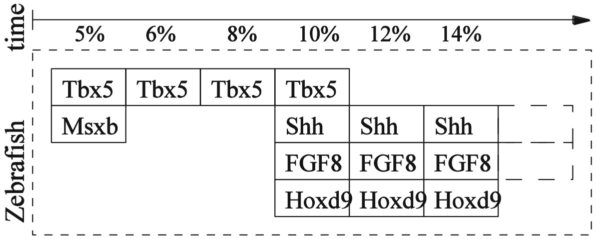


Fig. 7. Data recording of a selection of gene expression patterns in zebrafish in a relative time scale: *tbx5*, *msxb*, *ssh*, *fgf8*, *hoxb9*. At 10% of development 4 of the genes are expressed whereas at 14% development only 3 of the genes are expressed.

3.4 Sequences of Gene Expression in Development

Next, FEDA is applied to patterns of genes expression as found in the development of several model species. The clustering was performed with a window size of 4 and a minimal frequency of 0.04; the result corresponds with consensus in biological literature [14]. This result indicates that there is sufficient information in the data to differentiate between groups of species. The gene expression is analyzed to clades and therefore, subsequently, visualized as a cladogram. This cladogram is depicted in Figure 8.

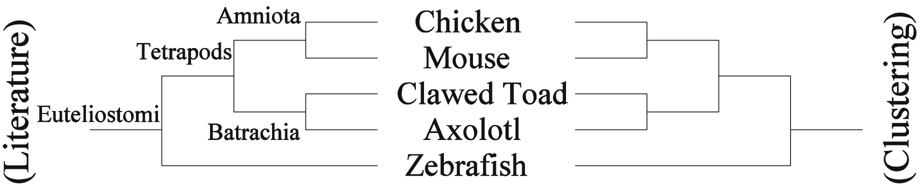


Fig. 8. Cladogram computed by FEDA based on gene-expression data (right) compared to a phylogenetic tree taken from biological literature [14] (left)

4 Conclusions and Discussion

We presented a method for the discovery of frequent patterns in a group of developmental sequences for quantitative analysis of heterochrony. All the episodes found in developmental sequences are found together with their frequency and a list of supporting sequences. These episodes are used in further analysis, such as clustering. Compared to previous experiments [4] our method is considerably more efficient. Furthermore, we demonstrated that transpositions of the data enable comparing morphological characters and genes as well as species in a transparent way. We have illustrated that our algorithm works with artificial as well as biological data

Currently, two methods are used for the analysis of developmental sequences of events, i.e. Event-pairing [18,11,16,17,8] and Search-Based Character Optimization [15]. Over Event-pairing [11] our method has two advantages. It uses the data to determine which pairs are the most interesting to use and the "event-pairs" can contain more than two events, so, in fact they are groups of developmental events that co-occur frequently. Groups of events found by FEDA contain more information about developmental sequences compared to event-pairs.

Search-Based Character Optimization [15] shows excellent clustering results and can possibly also be applied in the analysis of gene expression data. Over this method FEDA has two advantages. It allows insight in clustering, because FEDA is based on frequent developmental patterns and these patterns can later on be used to obtain more insight into which patterns cause the tree to branch. In addition, FEDA scales better to the size of the dataset and the number of events used in the analysis. FEDA is only based on patterns that are frequent, thus allowing it to handle large amount of data with a large number of developmental events. This does not restrict our method to sequences that contain all events because in sequences with missing events we are still able to find developmental patterns, just not the patterns that contain this missing event. Furthermore, our method does not use an edit cost matrix. For long developmental sequences with a large number of possible events this edit cost matrix extends to enormous and impractical proportions. Our method has the advantage that no step costs have to be determined, because the distances between species are only based on the data and the developmental patterns found.

The FEDA algorithm is developed to scale to larger datasets that will become available from genomics and developmental biology [5,6,7,19,20,21]. In that respect future directions for usage will extend beyond the analysis of heterochrony and include other aspects of computational evo-devo.

Acknowledgements

This work is partially supported the Netherlands' council for Scientific Research (NWO), grant #050.50.213 (Bio Molecular Informatics).

References

[1] http://www.r-project.org

[2] http://www.mysql.com

[3] http://www.ncbi.nlm.nih.gov/Taxonomy/CommonTree/wwwcmt.cgi

[4] Bathoorn, R., Siebes, A.: Constructing (Almost) Phylogenetic Trees from Developmental Sequences Data. In: 8th European Conf. on Principles and Practice of Knowledge Discovery in Databases, pp. 500–502 (2004)

[5] Belmamoune, M., Verbeek, F.J.: Heterogeneous Information Systems: bridging the gap of time and space. In: Management and retrieval of spatio-temporal Gene Expression data, InScit 2006, Merida, Spain (2006)

[6] Belmamoune, M., Verbeek, F.J.: Data Integration for Spatio-Temporal Patterns of Gene Expression of Zebrafish development: the GEMS database. J. of Integrative BioInformatics 5(2), 92 (2008)

[7] Belmamoune, M., Potikanond, D., Verbeek, F.J.: Mining and analysing spatio-temporal patterns of gene expression in an integrative database framework. J. of Integrative Bioinformatics 7(3), 128 (2010)

[8] Bininda-Emonds, O.R.P., Jefferey, J.E., Richardson, M.K.: Is sequence heterochrony an important evolutionary mechanism in mammals? J. of Mammalian Evolution 10(4), 335–361 (2003)

[9] Jaccard, P.: Nouvelles recherches sur la distribution florale. Bull Soc. Vaudoise Sci. Nat. 44, 223–227 (1908)

[10] Johnson, S.C.: Hierarchical Clustering Schemes. Psychometrika 2, 241–254 (1967)

[11] Jeffery, J.E., Bininda-Emonds, O.R.P., Coates, M.I., Richardson, M.K.: Analyzing evolutionary patterns in amniote embryonic development. Evolution & Development 4(4), 292–302 (2002)

[12] Jeffery, J.E., Richardson, M.K., Coates, M.I., Bininda-Emonds, O.R.P.: Analyzing Developmental Sequences within a Phylogenetic Framework. Systematic Biology 51(3), 478–491 (2002)

[13] Mannila, H., Toivonen, H., Verkamo, A.I.: Discovering frequent episodes in sequences. In: 1st Int. Conf. on Knowledge Discovery and Data Mining, pp. 210–215 (1995)

[14] Metscher, B.D., Ahlberg, P.E.: Zebrafish in Context: Use of a Laboratory Model in Comparative Studies. Develomental Biology 210, 1–14 (1999)

[15] Schulmeister, S., Wheeler, W.C.: Comparative and Phylogenetic analysis of developmental sequences. Evolution & Development 6(1), 50–57 (2004)

[16] Smith, K.K.: Sequence heterochrony and the evolution of development. Journal of morphology 252, 82–97 (2002)

[17] Smith, K.K.: Time's arrow: heterochrony and the evolution of development. Int. J. Dev. Biol. 47, 613–621 (2003)

[18] Schlosser, G.: Using heterochrony plots to detect the dissociated coevolution of characters. Journal of experimental zoology (mol dev evol) 291, 282–304 (2001)

[19] Verbeek, F.J., Lawson, K.A., Bard, J.B.L.: Developmental BioInformatics: linking genetic data to virtual embryos. Int. J. Dev. Biol. 43, 761–771 (1999)

[20] Verbeek, F.J., Rodrigues, D.D., Spaink, H., Siebes, A.: Data submission of 3D image sets to a bio-molecular database using active shape models and a 3D reference model for projection. In: Proceedings SPIE, Internet Imaging V, vol. 5304, pp. 13–23 (2004)

[21] Welten, M.C.M.: Spatio-temporal gene expression analysis from 3D in situ hybridisation images. PhD Thesis, Leiden University (2007)

Time Series Gene Expression Data Classification via L_1-norm Temporal SVM

Carlotta Orsenigo and Carlo Vercellis

Dept. of Management, Economics and Industrial Engineering,
Politecnico di Milano Via Lambruschini 4b, 20156 Milano, Italy
{carlotta.orsenigo,carlo.vercellis}@polimi.it

Abstract. Machine learning methods have been successfully applied to the phenotype classification of many diseases based on static gene expression measurements. More recently microarray data have been collected over time, making available datasets composed by time series of expression gene profiles. In this paper we propose a new method for time series classification, based on a temporal extension of L_1-norm support vector machines, that uses dynamic time warping distance for measuring time series similarity. This results in a mixed-integer optimization model which is solved by a sequential approximation algorithm. Computational tests performed on two benchmark datasets indicate the effectiveness of the proposed method compared to other techniques, and the general usefulness of the approaches based on dynamic time warping for labeling time series gene expression data.

Keywords: Time series classification, microarray data, L_1-norm support vector machines, dynamic time warping.

1 Introduction

In the last decade several machine learning methods have been proposed for the classification of gene expression data based on static datasets [1–4]. These datasets are usually composed by a huge number of features (genes) and a relatively few number of examples, and their values represent gene expression levels observed in a snapshot under precise experimental conditions.

The analysis of microarray expression levels recorded at a single time frame has proven to be effective for several biomedical tasks, among which the most prominent one is the phenotype classification in the early stages of a disease. However, it may appear inadequate to properly grasp the complex evolving interactions steering the biological processes. For example, in functional genomics studies the automatic categorization of genes based on their temporal evolution in the cell cycle plays a primary role, since genes with similar expression profiles are supposed to be functionally related or co-regulated [5]. As another example consider the prediction of the clinical response to a drug [6], where patients may exhibit different rates of disease development or treatment response. In this case, the overall profiles of the expression levels of two patients may be similar but not

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 264–274, 2010.

aligned, since individuals may progress at different speed [7]. In both scenarios it is required to analyze gene expression profiles as they evolve over time and, consequently, to develop classification methods able to consider also the temporal dimension. Over recent years a growing number of microarray experiments have been performed in order to collect and analyze time series gene expression profiles. The resulting datasets then provide examples of labeled time series that can be useful for classifying new temporal sequences whose label is unknown, and for identifying hidden explanatory biological patterns.

More generally, time series classification is a supervised learning problem aimed at labeling temporally structured univariate or multivariate sequences. Several alternative paradigms for time series classification have been proposed in the literature; see the review [8]. A common approach is based on a two-stage procedure that first derives a rectangular representation of the time series and then applies a classification method for labeling the data. An alternative approach relies on the notion of dynamic time warping (DTW) distance, an effective measure of similarity between pairs of time series. This distance allows to detect clusters and to predict with high accuracy the class of new temporal sequences by using distance-based methods, such as the k-nearest neighbor classifier [9, 10]. Furthermore, kernels based on DTW have been devised and incorporated within traditional support vector machines in [11–13].

In this paper we propose a new classification method based on a temporal variant of L_1-norm support vector machines (SVM), denoted as L_1-TSVM. The resulting mixed-integer optimization model, solved by a sequential approximation algorithm, takes into account the similarity among time series assigned to the same class, by including into the objective function a term that depends on the warping distances. A first research contribution along these lines is presented in [14], in which authors propose a temporal extension of discrete SVM, a variant of SVM based on the idea of accurately evaluating the number of misclassified examples instead of measuring their distance from the separating hyperplane [15, 16]. In this paper L_1-norm SVM [17–19] have been preferred as the base classifier for incorporating the temporal extension since they are efficient and well suited to deal with datasets with a high number of attributes, particularly in presence of redundant noisy features.

A second aim of the paper is to investigate whether DTW distance can be generally beneficial to different classifiers for labeling time series gene expression data. To this purpose, we comparatively evaluated the performances of five alternative methods beside L_1-TSVM: these are L_1-norm SVM, L_2-norm SVM with radial basis function and DTW as kernels, and the k-nearest neighbor (k-NN) classifier either based on Euclidean or DTW distances. Computational tests performed on two datasets seem to indicate that the proposed method L_1-TSVM has a great potential to perform an accurate classification of time series gene expression profiles and that, in general, SVM techniques based upon DTW perform rather well with respect to their non-DTW-based counterparts.

The paper is organized as follows. Section 2 defines time series classification problems and the concept of warping distance. In section 3 a new classification model based on L_1-norm temporal SVM is presented. In section 4 computational experiences are illustrated. Finally, section 5 discusses some future extensions.

2 Time Series Classification and Warping Distance

In a time series classification problem we are given a set of multivariate time series $\{\mathbf{A}_i\}$, $i \in \mathcal{M} = \{1, 2, \ldots, m\}$, where each $\mathbf{A}_i = [a_{ilt}]$ is a rectangular matrix of size $L \times T_i$ of real numbers. Here $l \in \mathcal{L} = \{1, 2, \ldots, L\}$ is the index associated to the *attributes* of the time series, whereas $t \in \mathcal{T}_i = \{1, 2, \ldots, T_i\}$ is the temporal index, that may vary in a different range for each $\mathbf{A}_i$. Every time series is also associated with a *class label* $y_i \in \mathcal{D}$. Let $\mathcal{H}$ denote a set of functions $f : \Re^n \mapsto \mathcal{D}$ that represent hypothetical relationships between $\{\mathbf{A}_i\}$ and y_i. The *time series classification problem* consists of defining an appropriate hypotheses space $\mathcal{H}$ and a function $f^* \in \mathcal{H}$ which optimally describes the relationship between the time series $\{\mathbf{A}_i\}$ and their labels $\{y_i\}$, in the sense of minimizing some measure of misclassification. When there are only two classes, i.e. $D = 2$ and $y_i \in \{-1, 1\}$ without loss of generality, we obtain a *binary* classification problem, while the general case is termed *multicategory* classification.

The *warping distance* has proven to be an effective proximity measure for clustering and labeling univariate time series [9, 10]. Indeed, it appears more robust than the Euclidean metric as a similarity measure, since it can handle sequences of variable length and automatically align the time series to identify similar profiles with different phases.

In order to find the optimal alignment between two time series $\mathbf{A}_i$ and $\mathbf{A}_k$, let $G = (V, E)$ be a directed graph whose vertices in V correspond to the pair of time periods $(r, s), r \in \mathcal{T}_i, s \in \mathcal{T}_k$. A vertex $v = (r, s)$ indicates that the r-th value of the time series $\mathbf{A}_i$ is matched with the s-th value of $\mathbf{A}_k$. An oriented arc (u, v) connects vertex $u = (p, q)$ to vertex $v = (r, s)$ if and only if one of the following mutually exclusive conditions holds

$$\{r = p + 1, s = q\} \vee \{r = p + 1, s = q + 1\} \vee \{r = p, s = q + 1\}. \tag{1}$$

Consequently, each vertex $u \in G$ has at most three outgoing arcs, associated to the three conditions described in (1). The arc (u, v) connecting the vertices $u = (p, q)$ and $v = (r, s)$ has length

$$\gamma_{uv} = \sum_{l=1}^{L} (a_{ilr} - a_{kls})^2, \tag{2}$$

given by the sum over the attributes of the squared distances associated to the potential alignment of period r in $\mathbf{A}_i$ to period s in $\mathbf{A}_k$. Let also $v_f = (1, 1)$ and $v_l = (T_i, T_k)$ be the vertices corresponding to the alignment of the first and last periods in the two sequences, respectively.

A *warping path* in G is any path connecting the source vertex v_f to the destination vertex v_l. It identifies a phasing and alignment between two time series such that matched time periods are monotonically spaced in time and

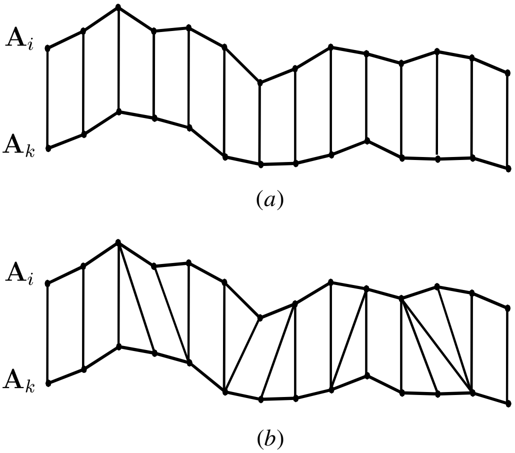


Fig. 1. Alignment of $\mathbf{A}_i$ and $\mathbf{A}_k$with Euclidean distance (a) and DTW distance (b)

contiguous. The *warping distance* between time series $\mathbf{A}_i$ and $\mathbf{A}_k$ is then defined as the length of the shortest warping path in G, and provides a measure of similarity between two temporal sequences which is often more effective than the Euclidean metric, as shown in Figure 1.

The warping distance between $\mathbf{A}_i$ and $\mathbf{A}_k$ can be evaluated by a dynamic optimization algorithm, with time complexity $O(T_{max}^2)$ ($T_{max} = \max\{T_i : i \in \mathcal{M}\}$), based on the following recursive equation

$$g(r,s) = \gamma_{uv} + \min\{g(r-1,s-1), g(r-1,s), g(r,s-1)\}, \tag{3}$$

where $g(r,s)$ denotes the cumulative distance of a warping path aligning the time series through the periods going from the pair $(1,1)$ to the pair (r,s).

3 L_1-norm Temporal Support Vector Machines

In this section we propose a new classification method based on a temporal variant of L_1-norm SVM, denoted as L_1-TSVM. The resulting mixed-integer optimization model, solved by a sequential approximation algorithm, takes into account the similarity among time series assigned to the same class, by including into the objective function a term that depends on the warping distances. We confine our attention to binary classification, since multicategory classification problems can be reduced to sequences of binary problems by means of *one-against-all* or *all-against-all* schemes [16, 20]. By applying an appropriate rectangularization preprocessing step, as described in section 4 for the time series considered in our tests, we may assume that the input dataset is represented by a $m \times n$ matrix, in which each row is a vector of real numbers $\mathbf{x}_i \in \Re^n$ which represents the corresponding time series $\mathbf{A}_i$.

A linear hypothesis for binary classification corresponds to a space $\mathcal{H}$ composed by separating hyperplanes taking the form $f(\mathbf{x}) = \operatorname{sgn}(\mathbf{w}'\mathbf{x} - b)$. In order to choose the optimal parameters $\mathbf{w}$ and b, traditional SVM [21–23], hereafter denoted as L_2-norm SVM, resort to the solution of the quadratic minimization problem

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{w}\|_2 + C\sum_{i=1}^{m}\xi_i && (L_2\text{-SVM}) \\ \text{s.t.} \quad & y_i\left(\mathbf{w}'\mathbf{x}_i - b\right) \geq 1 - \xi_i \quad i \in \mathcal{M} && (4) \\ & \xi_i \geq 0 \; \forall i; \quad \mathbf{w}, b \text{ free.} \end{aligned}$$

Here the L_2-norm $\|\mathbf{w}\|_2$ is a regularization term, aimed at maximizing the margin of separation, whereas the second term in the objective function is a loss function expressing the distance of the misclassified examples from the canonical hyperplane delimiting the correct halfspace. The parameter C is available for adjusting the trade-off between the two terms in the objective function of problem L_2-SVM.

The quadratic formulation L_2-SVM has some advantages, which contributed to its popularity. Among others, it admits fast solution algorithms and, through its dual problem, it allows to implicitly apply kernel transformations for deriving nonlinear separations in the original input space from linear separations obtained in a high-dimensional Hilbert space.

Yet, other norms $\|\mathbf{w}\|_p$ have been considered in the literature as alternative ways for expressing the margin maximization. In particular, linear formulations have attracted much attention [17–19] since they can benefit from the high efficiency of the solution algorithms for linear optimization problems. The linear counterpart of problem L_2-SVM is given by the optimization model

$$\begin{aligned} \min \quad & \|\mathbf{w}\|_1 + C\sum_{i=1}^{m}\xi_i && (L_1\text{-SVM}) \\ \text{s.t.} \quad & y_i\left(\mathbf{w}'\mathbf{x}_i - b\right) \geq 1 - \xi_i \quad i \in \mathcal{M} && (5) \\ & \xi_i \geq 0 \; \forall i; \quad \mathbf{w}, b \text{ free.} \end{aligned}$$

Although not suited to host the kernel transformations applicable to L_2-SVM, the linear problem L_1-SVM has proven even more effective to achieve an accurate separation directly into the input space, particularly when the number of attributes is high and there are noisy unnecessary features.

We propose an extension of problem L_1-SVM by defining a new term aimed at improving the discrimination capability when dealing with time series classification problems. This additional term is given by the sum of the warping distances between all pairs of time series assigned to the same class. By including this term into the objective function we aim at deriving a separating hyperplane which maximizes the overall similarity among time series lying in the same halfspace.

Let d_{ik} denote the warping distance between the pair of time series $(\mathbf{A}_i, \mathbf{A}_k)$. We have to introduce binary variables expressing the number of misclassified examples as

$$p_i = \begin{cases} 0 & \text{if } \mathbf{w}'\mathbf{x}_i - b \geq 1 \\ 1 & \text{otherwise} \end{cases} . \tag{6}$$

In order to determine the best separating hyperplane for time series classification, the following mixed-integer optimization problem L_1-TSVM, termed L_1*-norm temporal support vector machines*, can be formulated

$$\min \quad \sum_{j=1}^{n} u_j + C \sum_{i=1}^{m} \xi_i + \delta \sum_{i=1}^{m} \sum_{k=i+1}^{m} d_{ik} r_{ik} \tag{L_1-TSVM}$$

$$\text{s.t.} \quad y_i \left(\mathbf{w}'\mathbf{x}_i - b\right) \geq 1 - \xi_i \quad i \in \mathcal{M} \tag{7}$$

$$-u_j \leq w_j \leq u_j \quad j \in \mathcal{N} \tag{8}$$

$$\frac{1}{S} \xi_i \leq p_i \leq S \xi_i \quad i \in \mathcal{M} \tag{9}$$

$$-r_{ik} \leq y_i \left(2p_i - 1\right) + y_k \left(2p_k - 1\right) \leq r_{ik} \quad i, k \in \mathcal{M}, i < k \tag{10}$$

$$u_j, \xi_i, r_{ik} \geq 0 \ \forall i, j, k; \quad p_i \in \{0, 1\} \ \forall i; \quad \mathbf{w}, b \text{ free.}$$

Here S is a sufficiently large constant; C and δ the parameters to control the trade-off among the objective function terms. The family of continuous bounding variables $u_j, j \in \mathcal{N}$, and the constraints (8) are introduced in order to linearize the first term $\|\mathbf{w}\|_1$ in the objective function of problem L_1-SVM. Constraints (9) are required to enforce the relationship between the slack variables ξ_i and the binary misclassification variables p_i. Finally, the family of continuous bounding variables $r_{ik}, i, k \in \mathcal{M}$, together with the constraints (10), are needed to express in linear form via the third term the inclusion of the sum of the warping distances between the time series, as shown in [14].

For determining a feasible suboptimal solution to model L_1-TSVM, we propose the following approximation procedure based on a sequence of linear optimization (LO) problems. In what follows R-TSVM denotes the LO relaxation of model L_1-TSVM, and t is the iteration counter.

Procedure L_1-TSVM$_{\text{SLO}}$

1. Set $t = 0$ and consider the relaxation R-TSVM$_0$ of L_1-TSVM.
2. Solve problem R-TSVM$_t$.
3. Suppose first that problem R-TSVM$_t$ is feasible. If its optimal solution is integer, the procedure is stopped and the solution generated at iteration t is retained as an approximation to the optimal solution of L_1-TSVM; otherwise, proceed to step 5.
4. Otherwise, if problem R-TSVM$_t$ is unfeasible, modify previous problem R-TSVM$_{t-1}$ by fixing to 1 all of its fractional variables. Problem R-TSVM$_t$ redefined in this way is necessarily feasible and any of its optimal solutions is integer. Thus, the procedure is stopped and the solution found is retained as an approximation to the optimal solution of L_1-TSVM.

5. Next problem R-TSVM$_{t+1}$ in the sequence is obtained by fixing to zero the relaxed binary variable with the smallest fractional value in the optimal solution of the predecessor R-TSVM$_t$; then proceed to step 2.

4 Computational Experiments

Computational experiments were performed on two datasets both composed by microarray time series gene expression data. As stated in the introduction our aim was twofold; from one side, we intended to evaluate the effectiveness of L_1-TSVM and to compare it to its continuous counterpart in terms of accuracy. From the other side, we were interested in investigating whether DTW distance may be conveniently used in conjunction with alternative supervised learning methods for gene expression time series classification.

The first dataset considered in our tests, denoted as *Yeast*[1] and originally described in [24], contains the genome characterization of the mRNA transcript levels during the cell cycle of the yeast *Saccharomyces cerevisiae*. Gene expression levels were gathered at regular intervals during the cell cycle. In particular, measurements were performed at 17 time points with an interval of ten minutes between each pair of recorded values. The gene expression time series of this dataset are known to be associated to five different phases, namely Early G1, Late G1, S, G2 and M, which represent the class values in our setting. The second dataset, indicated as *MS-rIFN*β and first analyzed in [6], contains gene expression profiles of patients suffering from relapsing-remitting multiple sclerosis (MS), who are classified as either good or poor responders to recombinant human interferon beta (rIFNβ). The dataset is composed by the expression profiles of 70 genes isolated from each patient at 7 time points: before the administration of the first dose of the drug ($t = 0$), every three months ($t = 1, 2, 3, 4$) and every six months ($t = 5, 6$) in the first and second year of the therapy, respectively. For a few patients entire profile measurements are missing at one or two time points. From the complete *MS-rIFN*β dataset we retained only twelve genes whose expression profiles at $t = 0$ have shown to accurately predict the response to rIFNβ, as described in [6]. Furthermore, for each possible number of time points from 2 to 7 we extracted the corresponding gene expression time series, in order to obtain six different datasets. The distinctive features of *Yeast* and *MS-rIFN*β in terms of number of available examples, classes and time series length are summarized in Table 1.

Five alternative methods were considered for comparison with L_1-TSVM: L_1-SVM, SVM with radial basis function (SVM$_{\mathrm{RBF}}$) and dynamic time warping (SVM$_{\mathrm{DTW}}$) kernels, k-nearest neighbor classifier based respectively on Euclidean distance (k-NN$_{\mathrm{Eucl}}$) and dynamic warping distance (k-NN$_{\mathrm{DTW}}$). For solving L_1-TSVM and L_1-SVM models we respectively employed the heuristic procedure described in section 4 and standard LO code, both framed within the CPLEX environment; for nonlinear kernels SVM we used the LIBSVM library [25], extending its standard version with the DTW kernel. Among dynamic time

[1] http://genomics.stanford.edu/yeast_cell_cycle/cellcycle.html

Table 1. Summary of gene expression time series datasets

	Dataset	
Summary	*Yeast*	*MS-rIFNβ*
Examples	388	52
Classes	Early G1 (67), Late G1 (136), S (77) G2 (54), M (54)	Good responder (33), Poor responder (19)
Time series length	17	[5,7]

warping kernels we implemented the one proposed in [13], which has been proven to be positive definite under favorable conditions. Finally, in order to perform the classification of $Yeast$, which represents a multicategory dataset, SVM-based methods were framed within the all-against-all scheme.

A preprocessing step was applied on both datasets before classification. In particular, each expression profile of $Yeast$ was normalized as described in [24]. The expression levels of *MS-rIFNβ* were instead standardized, by subtracting from each value in a gene profile the mean of the values of the same gene in temporal-homologous sequences, and dividing the result by the corresponding standard deviation. Since all methods apart from SVM_{DTW} and k-NN_{DTW} are not able to cope with sequences of variable length, we replaced missing profiles with series of an out-of-range value, and then sequenced genes and time periods for every patient in order to obtain a rectangular representation for each of the six *MS-rIFNβ* datasets.

The accuracy of the competing methods was evaluated by applying five times 4-fold cross-validation, each time randomly dividing the dataset into four folds for training and testing. To achieve a fair comparison we used the same folds for all methods. Furthermore, on each training set we applied 3-fold cross-validation in order to figure out the optimal parameters setting for all classifiers, represented by the regularization constant C and the kernel parameter σ for L_1-norm and L_2-norm SVM methods, and by the number k of neighbors for k-NN classifiers. For L_1-TSVM a further parameter to be optimized was represented by the weight δ in the objective function, regulating the trade-off between misclassification and the sum of time series warping distances. The values tested for each parameter are reported in Table 2.

The results of each method are shown in Table 3 which indicates the average accuracy values obtained by applying five times 4-fold cross-validation. These results allow us to draw some empirical conclusions concerning the effectiveness of the proposed method and the usefulness of DTW distance. The temporal variant L_1-TSVM was capable of outperforming its counterpart L_1-SVM on all datasets, achieving an increase in accuracy ranging between 0.8% and 5.4%. The novel technique appeared rather accurate also with respect to the other classifiers, being able to provide the highest rate of correct predictions on $Yeast$ and on most *MS-rIFNβ* datasets. Especially on these datasets, in which

Table 2. Parameters values tested for each family of methods

Method	Parameters values
k-NN$_{\text{Eucl}}$ k-NN$_{\text{DTW}}$	$\text{k} = 2, 4, 6, 8, 10$
SVM$_{\text{RBF}}$ SVM$_{\text{DTW}}$	$\text{C} = 10^j, j \in [-1, 3]$ $\sigma = 10^j, j \in [-4, 2]$
L_1-SVM L_1-TSVM	$\text{C} = 10^j, j \in [-1, 3]$ $\delta = 10^j, j \in [-1, 1]$

Table 3. Classification accuracy (%) on the gene expression time series datasets. Intervals in brackets indicate the time points considered for each *MS-rIFNβ* dataset.

	Method					
Dataset	k-NN$_{\text{Eucl}}$	k-NN$_{\text{DTW}}$	SVM$_{\text{RBF}}$	SVM$_{\text{DTW}}$	L_1-SVM	L_1-TSVM
Yeast	68.5	51.8	73.3	73.7	72.4	**73.9**
MS-rIFNβ						
t∈[0,1]	83.8	76.9	82.7	**84.2**	76.9	80.8
t∈[0,2]	81.9	78.9	82.7	84.6	80.0	**85.4**
t∈[0,3]	82.7	75.0	81.9	75.4	78.5	**83.8**
t∈[0,4]	76.9	73.1	76.9	71.2	79.2	**80.0**
t∈[0,5]	75.8	69.2	71.5	78.5	79.6	**80.8**
t∈[0,6]	71.2	66.9	68.5	70.8	76.5	**78.8**

examples are composed by sequences of variable length, also the use of DTW as the kernel function appeared promising. Notice that the average accuracy provided by most classifiers on *MS-rIFNβ* datasets decreased when more than four expression time series were considered for each example. This phenomenon is possibly related to the increase of missing profiles in the last measurements which may have slightly compromised the classification results. Nevertheless, L_1-TSVM showed the mildest degradation of its classification performances with respect to most of the other methods. On the *Yeast* dataset the competing techniques L_1-TSVM and SVM$_{\text{DTW}}$ provided comparable results. Even in this case, however, L_1-TSVM was able to obtain the best rate of correct predictions. By investigating the confusion matrices of all methods we observed that the higher accuracy of L_1-TSVM mainly derived from the correct classification of a greater number of examples belonging to the classes S and M.

5 Conclusions and Future Extensions

In this paper we have proposed a new supervised learning method for time series gene expression classification based on a temporal extension of L_1-norm SVM.

The novel technique relies on a mixed-integer optimization problem which incorporates in the objective function an additional term aiming at improving the discrimination capability when dealing with the classification of time series datasets. This term is represented by the sum of the warping distances of time series assigned to the same class, where the warping distance is used as a similarity measure among temporal sequences. The inclusion of this term in the objective function is aimed at deriving separating hyperplanes which are also optimal with respect to time series similarity. In this paper we have also investigated from a computational perspective the convenience of combining the warping distance with alternative classification methods for time series gene profiles labeling. Experiments performed on two datasets showed the effectiveness of the proposed method and the usefulness of the warping distance when used as the kernel function in L_2-norm SVM. Future research development will be pursued along three main directions, by testing the novel technique on a wider range of time series gene expression classification problems, by investigating other similarity measures to be included in the model and by studying alternative heuristic procedures for solving the resulting mixed-integer formulations.

References

1. 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.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286, 531–537 (1999)
2. Furey, T., Cristianini, N., Duffy, N., Bednarski, D., Schummer, M., Haussler, D.: Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16, 906–914 (2000)
3. Lai, C., Reinders, M.J.T., van't Veer, L.J., Wessels, L.F.A.: A comparison of univariate and multivariate gene selection techniques for classification of cancer datasets. BMC Bioinformatics 7, 235 (2006)
4. Cho, S.B., Won, H.H.: Cancer classification using ensemble of neural networks with multiple significant gene subsets. Applied Intelligence 26, 243–250 (2007)
5. Peddada, S., Lobenhofer, E., Li, L., Afshari, C., Weinberg, C., Umbach, D.: Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference. Bioinformatics 19, 834–841 (2003)
6. Baranzini, S., Mousavi, P., Rio, J., Caillier, S., Stillman, A., Villoslada, P., Wyatt, M., Comabella, M., Greller, L., Somogyi, R., Montalban, X., Oksenberg, J.: Transcription-based prediction of response to IFNβ using supervised computational methods. PLoS Biology 3, 166–176 (2005)
7. Lin, T., Kaminski, N., Bar-Joseph, Z.: Alignment and classification of time series gene expression in clinical studies. In: ISMB (Supplement of Bioinformatics), pp. 147–155 (2008)
8. Kadous, M.W., Sammut, C.: Classification of multivariate time series and structured data using constructive induction. Machine Learning 58, 179–216 (2005)
9. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. Knowledge and Information Systems 7, 358–386 (2004)
10. Xi, X., Keogh, E., Shelton, C., Wei, L.: Fast time series classification using numerosity reduction. In: Proc. of the 23rd International Conference on Machine Learning, pp. 1033–1040 (2006)

11. Shimodaira, H., Noma, K.I., Nakai, M., Sagayama, S.: Dynamic time-alignment kernel in support vector machine. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) NIPS, pp. 921–928. MIT Press, Cambridge (2001)
12. Bahlmann, C., Haasdonk, B., Burkhardt, H.: On-line handwriting recognition with support vector machines: A kernel approach. In: IWFHR '02: Proc. of the Eighth International Workshop on Frontiers in Handwriting Recognition, pp. 49–54. IEEE Computer Society, Washington (2002)
13. Cuturi, M., Vert, J.P., Birkenes, O., Matsui, T.: A kernel for time series based on global alignments. In: Proc. of ICASSP, pp. 413–416 (2007)
14. Orsenigo, C., Vercellis, C.: Combining discrete SVM and fixed cardinality warping distances for multivariate time series classification. Pattern Recognition 43, 3787–3794 (2010)
15. Orsenigo, C., Vercellis, C.: Discrete support vector decision trees via tabu-search. Journal of Computational Statistics and Data Analysis 47, 311–322 (2004)
16. Orsenigo, C., Vercellis, C.: Multicategory classification via discrete support vector machines. Computational Management Science 6, 101–114 (2009)
17. Bradley, P.S., Mangasarian, O.L.: Massive data discrimination via linear support vector machines. Optimization Methods and Software 13, 1–10 (2000)
18. Zhu, J., Rosset, S., Hastie, T., Tibshirani, R.: 1-norm support vector machines. Neural Information Processing Systems 16 (2003)
19. Mangasarian, O.L.: Exact 1-norm support vector machines via unconstrained convex differentiable minimization. Journal of Machine Learning Research 7, 1517–1530 (2006)
20. Allwein, E., Schapire, R., Singer, Y.: Reducing multiclass to binary: a unifying approach for margin classifiers. Journal of Machine Learning Research 1, 113–141 (2000)
21. Vapnik, V.: The nature of statistical learning theory. Springer, New York (1995)
22. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge (2000)
23. Schölkopf, B., Smola, A.: Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT Press, Cambridge (2002)
24. Cho, R.J., Campbell, M.J., Winzeler, E.A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T.G., Gabrielian, A.E., Landsman, D., Lockhart, D.J., Davis, R.W.: A genome-wide transcriptional analysis of the mitotic cell cycle. Molecular Cell 2, 65–73 (1998)
25. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm

Part IV

Bioimaging

Sub-grid and Spot Detection in DNA Microarray Images Using Optimal Multi-level Thresholding

Iman Rezaeian and Luis Rueda

School of Computer Science, University of Windsor
401 Sunset Ave., Windsor, ON, N9B3P4, Canada
{rezaeia,lrueda}@uwindsor.ca

Abstract. The analysis of DNA microarray images is a crucial step in gene expression analysis, since any errors in early stages are propagated in future steps in the analysis. When processing the underlying images, accurately separating the sub-grids and spots is of extreme importance for subsequent steps that include segmentation, quantification, normalization and clustering. We propose a fully automatic approach that first detects the sub-grids given the entire microarray image, and then detects the locations of the spots in each sub-grid. The approach first detects and corrects rotations in the images by an affine transformation, followed by a polynomial-time optimal multi-level thresholding algorithm to find the positions of the sub-grids and spots. Additionally, a new validity index is proposed in order to find the correct number of sub-grids in the microarray image, and the correct number of spots in each sub-grid. Extensive experiments on real-life microarray images show that the method performs these tasks automatically and with a high degree of accuracy.

Keywords: Microarray image gridding; image analysis; multi level thresholding.

1 Introduction

Microarrays are one of the most important technologies used in molecular biology to massively explore the abilities of the genes to express themselves into proteins and other molecular machines responsible for different functions in an organism. These expressions are monitored in cells and organisms under specific conditions, and are present in many applications in medical diagnosis, pharmacology, disease treatment, just to mention a few. We consider DNA microarrays, which are produced on a slide, typically, in two channels. Scanning the slides at a very high resolution produces images composed of sub-grids of spots. Image processing and analysis are two important aspects of microarrays, since the aim of the whole experimental procedure is to obtain meaningful biological conclusions, which depends on the accuracy of the different stages, mainly those at the beginning of the process. The first task is gridding, which if done correctly, helps substantially improve the efficiency of the subsequent steps that include segmentation, quantification, normalization and data mining. When producing DNA microarrays, many parameters are specified, such as the number and size of spots, number of sub-grids, and even their exact location. However, many physicochemical factors

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 277–288, 2010.

produce noise, misalignment, and even deformations in the sub-grid template that it is virtually impossible to know the exact location of the spots after the scanning is performed, at least with the current technology. Roughly speaking, gridding consists of determining the spot locations in a microarray image (typically, in a sub-grid). The gridding process requires the knowledge of the sub-girds in advance in order to proceed.

Many approaches have been proposed for sub-gridding and spot detection. The Markov random field (MRF) is a well known approach that applies specific constraints and heuristic criteria [11]. Another gridding method is mathematical morphology, which represents the image as a function and applies erosion operators and morphological filters, helping remove peaks and ridges from the topological surface of the images [7]. A method for detecting spot locations based on a Bayesian model has been recently proposed, and uses a deformable template model to fit the grid of spots in such a template using a posterior probability model which learns its parameters by means of a simulated-annealing-based algorithm [1,6]. Another method for finding spot locations uses a hill-climbing approach to maximize the energy, seen as the intensities of the spots which are fit to different probabilistic models [10]. Fitting the image to a mixture of Gaussians is another technique that has been applied to gridding microarray images by considering radial and perspective distortions [5]. A Radon-transform-based method that separates the sub-grids in a DNA microarray image has been proposed in [8]. Other approaches for DNA microarray gridding include the following. A gridding method that performs sub-gridding and spot detection is the one proposed in [13], which performs a series of steps including rotation detection based on a simple method that compares the running sum of the topmost and bottommost parts of the image. This method, which detects rotation angles with respect to one of the axes, either x or y, has not been tested on images having regions with high noise (e.g. bottommost $\frac{1}{3}$ of the image is quite noisy). Another method for gridding DNA microarray images uses an evolutionary algorithm to separate sub-grids and detect the positions of the spots [4]. The approach is based on a genetic algorithm that discovers parallel and equidistant line segments that compose the grid structure. Using maximum margin is another method for automatic gridding of DNA microarray images based on the maximization of the margin between the rows and columns of the spots [2]. In another approach, properties of planar (2D) grids are addressed from a mathematical point of view and an algorithm for recognizing distorted grids with perspective transformations is presented [5]. The approach involves recognizing parameters of affinely distorted grids by fitting Gaussian mixture models to grid spectrums, rebuilding the grid structures via a generating iteration based on the acquired parameters, and eliminating nonlinear effects caused by perspective transformations with the median of infinite lines from local structures.

In this paper, we propose a fully automatic approach that first detects the sub-grids given the entire microarray image, and then detects the locations of the spots in each sub-grid. The method proposed here uses an optimal multi-level thresholding algorithm to find the positions of the sub-grids in the image and the positions of the spots in each sub-grid. Additionally, a new validity index is proposed in order to find the correct number of sub-grids in the microarray image, and the correct number of spots in each sub-grid.

2 The Proposed Gridding Method

A DNA microarray image typically contains a number of sub-grids and each sub-grid contains a number of spots arranged in rows and columns. The aim is to perform a two-stage process in such a way that the sub-grid locations are found in first stage, and then spot locations within a sub-grid can be found in the second stage. Consider an image (matrix) $A = \{a_{ij}\}, i = 1, ..., n$ and $j = 1, ..., m$, where $a_{ij} \in \mathbb{Z}^+$ and a_{ij} represents the intensity of pixel (i,j) (usually, a_{ij} is in the range [0..65,535] in a TIFF image). The aim of gridding is to obtain a matrix G (grid) where $G = \{g_{ij}\}, i = 1, ..., n$ and $j = 1, ..., m$, $g_{ij} = 0$ or $g_{ij} = 1$ (a binary image), with 0 meaning that g_{ij} belongs to a grid separator, and 1 meaning the pixel is inside a spot region. This image could be thought of as a "free-form" grid. However, in order to restrict our definition to a rectangular grid, our aim is to obtain vectors **v** and **h**, **v** = $[v_1, ...v_m]^t$, **h** = $[h_1, ...h_n]^t$, where $v_i \in [1, m]$ and $h_j \in [1, n]$. Each vertical and horizontal vectors are used to separate sub-grids and spots.

The sub-grids in a microarray image are detected by applying the Radon transform as a preprocessing phase and then using optimal multilevel thresholding in the next stage. By combining optimal multilevel thresholding and the β index (Eq. 13), the correct number of thresholds (sub-grids or spots) can be found. Figure 1 depicts the process of finding the sub-grids in a microarray image. The input for the Radon transform process is a microarray image and the output of the whole process is the location (and partitioning) of the sub-grids. Analogously, the locations of the spots in each sub-grid are found by using optimal multilevel thresholding combined with the proposed β index to find the best number of rows and columns of spots. The input for this process is a sub-grid (already extracted from the sub-gridding step) and the output is given by the partitioning of the sub-grid into spots (spot regions).

We apply the Radon transform as a preprocessing step (to the raw images) in order to detect and correct rotations, if any, in the whole image or in a sub-grid. Rotations of an image can be seen in two different directions, with respect to the x and y axes. The aim is find two independent angles of rotation for an affine transformation, and for this the Radon transform is applied. Details on the Radon transform and how to use it in correcting rotations can be found in [8].

3 Optimal Multilevel Thresholding

Multilevel thresholding is one of the most widely-used techniques in image processing, including segmentation, classification and object discrimination. Given a histogram with frequencies or probabilities for each bin, the aim of multilevel thresholding is to divide the histogram into a number of groups (or classes) of contiguous bins in such a way that a criterion is optimized. In microarray image gridding, we compute the vertical (or horizontal) running sum of pixel intensities, obtaining a histogram in which each bin represents one column (or row respectively), and the running sum of intensities correspond to the frequency of that bin. The frequencies are then normalized in order to be considered as probabilities. Figure 2 depicts a typical DNA microarray image (AT-20387-ch2, see its description in section 5) that contains 12×4 sub-grids, along with

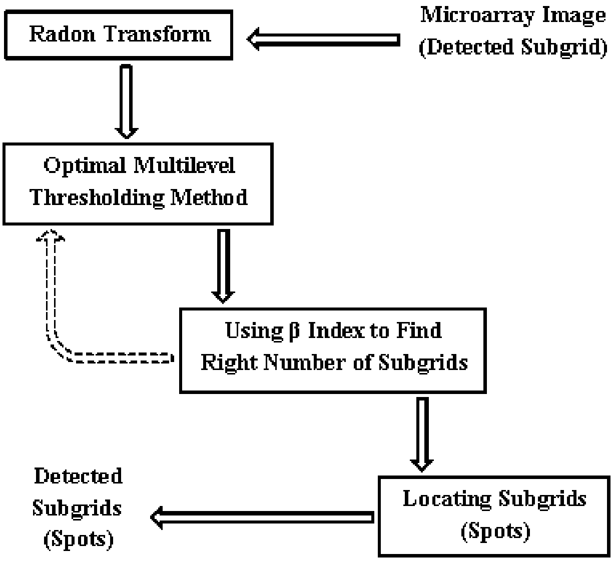

Radon Transform
Microarray Image (Detected Subgrid)
Optimal Multilevel Thresholding Method
Using β Index to Find Right Number of Subgrids
Locating Subgrids (Spots)
Detected Subgrids (Spots)

Fig. 1. Schematic representation of the process for finding sub-grids (spots) in microarray images (detected sub-grids)

the corresponding histograms representing the horizontal and vertical running sums. Each histogram is then processed (see below) to obtain the optimal thresholding that will determine the locations of the lines separating the sub-grids. Analogously, we apply the same method to each sub-grid to obtain the corresponding lines separating the spot regions.

Consider a histogram H, an ordered set $\{1, 2, \ldots, n-1, n\}$, where the ith value corresponds to the ith bin and has a probability, p_i. Given an image, $A = \{a_{ij}\}$, H can be obtained by means of the horizontal (vertical) running sum as follows: $p_i = \sum_{j=1}^{m} a_{ij}$ $(p_j = \sum_{i=1}^{n} a_{ij})$. We also consider a threshold set T, defined as an ordered set $T = \{t_0, t_1, \ldots, t_k, t_{k+1}\}$, where $0 = t_0 < t_1 < \ldots < t_k < t_{k+1} = n$ and $t_i \in \{0\} \cup H$. The problem of multilevel thresholding consists of finding a threshold set, T^*, in such a way that a function $f : H^k \times [0,1]^n \rightarrow \mathbb{R}^+$ is maximized/minimized. Using this threshold set, H is divided into $k+1$ classes: $\zeta_1 = \{1, 2, \ldots, t_1\}$, $\zeta_2 = \{t_1+1, t_1+2, \ldots, t_2\}, \ldots$, $\zeta_k = \{t_{k-1}+1, t_{k-1}+2, \ldots, t_k\}$, $\zeta_{k+1} = \{t_k+1, t_k+2, \ldots, n\}$. The most important criteria for multilevel thresholding are the following [9]:

Between class variance:

$$\Psi_{\mathrm{BC}}(T) = \sum_{j=1}^{k+1} \omega_j \mu_j^2 \tag{1}$$

where $\omega_j = \sum_{i=t_{j-1}+1}^{t_j} p_i$, $\mu_j = \frac{1}{\omega_j} \sum_{i=t_{j-1}+1}^{t_j} i p_i$;

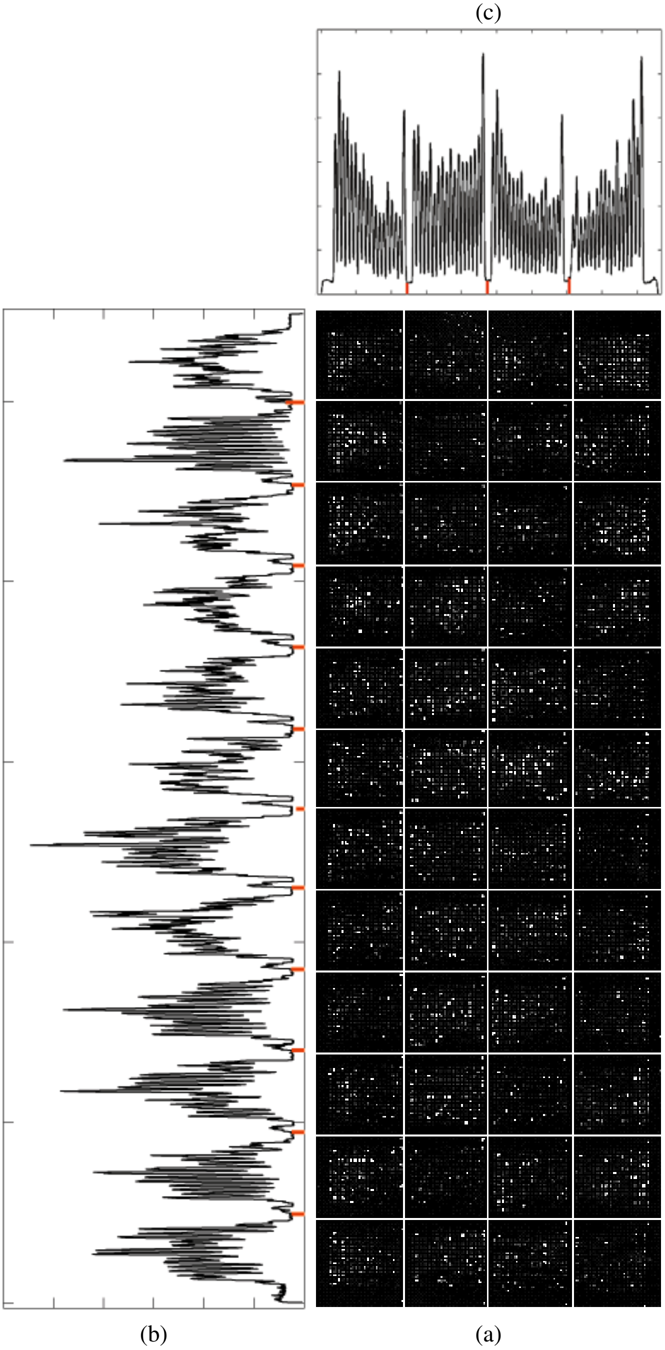


Fig. 2. (a) detected sub-grids in AT-20387-ch2 microarray image, (b) vertical histogram and detected valleys correspond to vertical lines, (c) horizontal histogram and detected valleys correspond to horizontal lines

Entropy-based:

$$\mathit{\Psi}_{\mathrm{H}}(T) = \sum_{j=1}^{k+1} H_j \tag{2}$$

where $H_j = -\sum_{i=t_{j-1}+1}^{t_j} \frac{p_i}{\omega_j} \log \frac{p_i}{\omega_j}$;

Minimum error:

$$\mathit{\Psi}_{\mathrm{ME}}(T) = 1 + 2\sum_{j=1}^{k+1} \omega_j(\log \sigma_j - \log \omega_j) \tag{3}$$

where $\sigma_j^2 = \sum_{i=t_{j-1}+1}^{t_j} \frac{p_i(i-\mu_j)^2}{\omega_j}$.

A dynamic programming algorithm for *optimal* multilevel thresholding was proposed in our previous work [9], which is an extension for irregularly sampled histograms. For this, the criterion has to be decomposed as a sum of terms as follows:

$$\mathit{\Psi}(T_{0,m}) = \mathit{\Psi}(\{t_0, t_1, \dots, t_m\}) \triangleq \sum_{j=1}^{m} \psi_{t_{j-1}+1,t_j} \, , \tag{4}$$

where $1 \le m \le k+1$ and the function $\psi_{l,r}$, where $l \le r$, is a real, positive function of $p_l, p_{l+1}, \dots, p_r$, $\psi_{l,r} \,:\, H^2 \times [0,1]^{l-r+1} \to \mathbb{R}^+ \cup \{0\}$. If $m = 0$, then $\mathit{\Psi}(\{t_0\}) = \psi_{t_0,t_0} = \psi_{0,0} = 0$. The thresholding algorithm can be found in [9]. In the algorithm, a table C is filled in, where $C(t_j, j)$ contains the optimal solution for $T_{0,j} = t_0, t_1, \dots, t_j$, $\mathit{\Psi}^*(T_{0,j})$, which is found from $\min\{t_j\} \le t_j \le \max\{t_j\}$. Another table, $D(t_j, j)$, contains the value of t_{j-1} for which $\mathit{\Psi}^*(T_{0,j})$ is optimal. The worst-case time complexity of the algorithm has been shown to be $\mathit{\Theta}(kn^2)$.

To implement the between-class variance criterion, the function $\mathit{\Psi}_{\mathrm{BC}}(T)$ is expressed as: $\mathit{\Psi}_{\mathrm{BC}}(T) = \sum_{j=1}^{k+1} \omega_j \mu_j^2 = \sum_{j=1}^{k+1} \psi_{t_{j-1}+1,t_j}$, where $\psi_{t_j+1,t_{j+1}} = \omega_j \mu_j^2$. We consider the temporary variables, a and b ,which are computed as follows:

$$a \leftarrow p_{t_{j-1}+1} + \sum_{i=t_{j-1}+2}^{t_j} p_i, \qquad \text{and} \tag{5}$$

$$b \leftarrow (t_{j-1}+1)p_{t_{j-1}+1} + \sum_{i=t_{j-1}+2}^{t_j} ip_i \, . \tag{6}$$

Since from (5) and (6), a and b are known, then $\psi_{t_{j-1}+2,t_j}$, for the next step, can be re-computed as follows in $\mathit{\Theta}(1)$:

$$a \leftarrow a - p_{t_{j-1}+1} \, , \tag{7}$$

$$b \leftarrow b - (t_{j-1}+1)p_{t_{j-1}+1} \, , \text{ and} \tag{8}$$

$$\psi_{t_{j-1}+2,t_j} \leftarrow \frac{b^2}{a} \, . \tag{9}$$

Similar decomposition allows that the minimum error and entropy-based criteria be recomputed in $\mathit{\Theta}(1)$ [9].

4 Automatic Detection of the Number of Sub-grids and Spots

Finding the correct number of sub-grids in a microarray image and number of spots in each sub-grid is one of the most important phases in sub-grid and spot detection. For this, we resort on validity indices used for clustering. By analyzing the traditional indices for clustering validity, we found that combining these indices with one of our measures, we propose a new index of validity for this specific problem. Initially, we considered four clustering validity indices (cf. [12]) in the context of the partitioning obtained by the multilevel thresholding method. We found that the best is the I index, which is defined as follows:

$$I(K) = \left(\frac{1}{K} \times \frac{E_1}{E_K} \times D_K \right)^2 , \tag{10}$$

where, $E_K = \Sigma_{i=1}^{K} \Sigma_{k=1}^{n_i} p_k ||k - z_i||$, $D_K = \underbrace{\overset{K}{max}}_{i,j=1} ||z_i - z_j||$, n is the total number of points in data set, and z_k is the center of the kth cluster. To find the best number of thresholds, we perform an exhaustive search on all possible values of K, from 2 to $\sqrt{n}$ [3]. The value of K for which $I(K)$ is maximal is considered to be correct number of clusters. We must note that the complexity of the algorithm remains $\Theta(kn^2)$, since the index I is computed for all values of K as the optimal thresholding algorithm fills in table C.

Based on this index and another measurement (see Eq. (11) below), we propose a new index for finding the correct number of sub-grids or spots. We consider the average value of the thresholds in a histogram, which is computed as follows:

$$A(K) = \frac{1}{K} \sum_{i=1}^{K} f(t_i), \tag{11}$$

where t_i is the ith threshold found by optimal multilevel thresholding and $f(t_i)$ is its respective value in the histogram.

The proposed index computes the value of A for different numbers of thresholds, K. Then, the best number of thresholds K^* can be found as follows:

$$K^* = arg \underbrace{min}_{1 \leq K \leq \delta} \left\{ \frac{1}{K} \Sigma_{i=1}^{K} f(t_i) \right\} \tag{12}$$

where δ is the maximum number of thresholds and equals to $\sqrt{n}$ [3]. Based on our experimental studies, the best results were obtained from a combination of our proposed index (11) and the I index (10) is as follows:

$$\beta(K) = \frac{I(K)}{A(K)} \tag{13}$$

For maximizing $I(K)$ and minimizing $A(K)$, the value of $\beta(K)$ must be maximized. Thus, the best number of thresholds K^* based on the β index is given by:

$$K^* = arg \underbrace{max}_{1 \leq K \leq \delta} \beta(K) = arg \underbrace{max}_{1 \leq K \leq \delta} \frac{\left(\frac{1}{K} \times \frac{E_1}{E_K} \times D_K \right)^2}{\frac{1}{K} \Sigma_{i=1}^{K} f(t_i)} \tag{14}$$

5 Experimental Results

For the experiments, two different kinds of cDNA microarray images have been used. The images have been selected from different sources, and have different scanning resolutions, in order to study the flexibility of the proposed method to detect sub-grids and spots with different sizes and features. The first set of images has been drawn from the Stanford Microarray Database (SMD), and corresponds to a study of the global transcriptional factors for hormone treatment of *Arabidopsis thaliana*[1] samples. Ten images were selected for testing the proposed method, and they correspond to channels 1 and 2 for experiments IDs 20385, 20387, 20391, 20392 and 20395. The images have been named using AT (which stands for *Arabidopsis thaliana*), followed by the experiment ID, and the channel number (1 or 2). The images have a resolution of 1910×5550 pixels and are in TIFF format. The spot resolution is 24×24 pixels per spot. Also, each image contains 48 sub-grids, arranged in 12 rows and 4 columns. The second test suite consists of a set of images from Gene Expression Omnibus (GEO) and corresponds to an Atlantic salmon macrophage study [2] samples. Eight images were selected for testing the proposed method, and they correspond to channels 1 and 2 for experiments IDs GSM16101, GSM16389 and GSM16391 and also channels 1 of GSM15898 and channels 2 of GSM15898. The images have been named using GSM followed by the experiment ID, and the channel number (1 or 2). The images have a resolution of 1900×5500 pixels and are in TIFF format. The spot resolution is 12×12 pixels per spot. Also each image contains 48 sub-grids, arranged in 12 rows and 4 columns.

To assess the performance of the detection method, we consider the following. We call *false positive* (FP) a grid line that separates an area into two different sub-areas and at least one of them does not contain a spot or a sub-grid. Similarly, a *false negative* (FN) occurs when two adjacent areas containing spots or sub-grids are not separated by a grid line. True positives (TP) and true negatives (TN) are obtained by the corresponding differences between the total number of cases minus FP or FN, respectively. Considering N as the total number of grid lines in the image, accuracy is calculated as $\frac{(TP+TN)}{N}$.

We have used the between-class variance as the thresholding criteria, since it is the one that delivers the best results. Table 1 shows the results of applying the proposed method for sub-grid and spot detection on the Thaliana dataset. All the sub-grids in each image are detected accurately, and also spot locations in each sub-grid can be detected efficiently with an average accuracy of 96.2% for this dataset. The same sets of experiments were repeated for the GEO dataset and the results are shown in Table 2. The sub-grids in each microarray image are accurately detected with a 100% accuracy and the spot locations in each sub-grid are detected efficiently with an average performance of 96% for this dataset. As shown in Tables 1 and 2, for all of images, in the sub-grid detection phase, the false negative and false positive rates are both 0%, yielding an accuracy of 100%. This means the proposed method works perfectly in sub-grid

[1] The images can be downloaded from smd.stanford.edu, by searching “Hormone treatment” as category and “Transcription factors” as subcategory.

[2] The images can be downloaded from ncbi.nlm.nih.gov, by selecting “GEO Datasets” as category and searching the name of images.

Table 1. Accuracy results of detected sub-grids and spots for each image in the Thaliana dataset and their respective FP and FN rates

	Sub-grid Detection			Spot Detection		
Image	False Negative	False Positive	Accuracy	False Negative	False Positive	Accuracy
AT-20385-CH1	0.0%	0.0%	100%	6.5%	0.4%	93.1%
AT-20385-CH2	0.0%	0.0%	100%	3.3%	1.5%	95.4%
AT-20387-CH1	0.0%	0.0%	100%	7.4%	0.5%	92.1%
AT-20387-CH2	0.0%	0.0%	100%	0.0%	0.6%	99.4%
AT-20391-CH1	0.0%	0.0%	100%	0.0%	1.2%	98.8%
AT-20391-CH2	0.0%	0.0%	100%	3.7%	1.3%	98.8%
AT-20392-CH1	0.0%	0.0%	100%	0.7%	1.0%	95.0%
AT-20392-CH2	0.0%	0.0%	100%	3.1%	1.3%	98.3%
AT-20395-CH1	0.0%	0.0%	100%	6.5%	0.4%	95.6%
AT-20395-CH2	0.0%	0.0%	100%	6.5%	0.4%	95.7%

Table 2. Accuracy results of detected sub-grids and spots for each image in the GEO dataset and their respective FP and FN rates

	Sub-grid Detection			Spot Detection		
Image	False Negative	False Positive	Accuracy	False Negative	False Positive	Accuracy
GSM15898-CH1	0.0%	0.0%	100%	3.2%	0.1%	96.7%
GSM15899-CH2	0.0%	0.0%	100%	3.2%	0.2%	96.6%
GSM16101-CH1	0.0%	0.0%	100%	3.0%	0.0%	97.0%
GSM16101-CH2	0.0%	0.0%	100%	3.1%	0.0%	96.9%
GSM16389-CH1	0.0%	0.0%	100%	5.8%	0.0%	94.2%
GSM16389-CH2	0.0%	0.0%	100%	3.1%	0.0%	96.9%
GSM16391-CH1	0.0%	0.0%	100%	6.7%	0.0%	93.3%
GSM16391-CH2	0.0%	0.0%	100%	3.6%	0.0%	96.4%

detection. Additionally, in the spot detection phase, accuracy of the proposed method is very high, being above 96% in both cases.

One of the reasons for the slightly lower accuracy in spot detection is that the distance between spots is smaller than the distance between sub-grids. In both datasets, there are approximately eight pixels between adjacent spots, and approximately 30 pixels horizontally and 100 pixels vertically between sub-grids in the Thaliana dataset, and 200 pixels between sub-grids in the GEO dataset. Another possible reason for this behavior is that the number of pixels in each sub-grid is far lower than a microarray image (around $\frac{1}{50}$). Thus, existing noise affects the spot detection phase much more than the noise present in the sub-grid extraction stage. It is important to highlight, however, that the sub-grid detection process is not affected by the presence of noise at all.

We have also performed a visual analysis in order to obtain a different perspective of our results. Figure 3 shows the detected sub-grids from the AT-20387-ch2 image (left) and the detected spots in one of sub-grids (right). As shown in the figure, the proposed method finely detects the sub-grids location at first, and in the next stage, each sub-grid

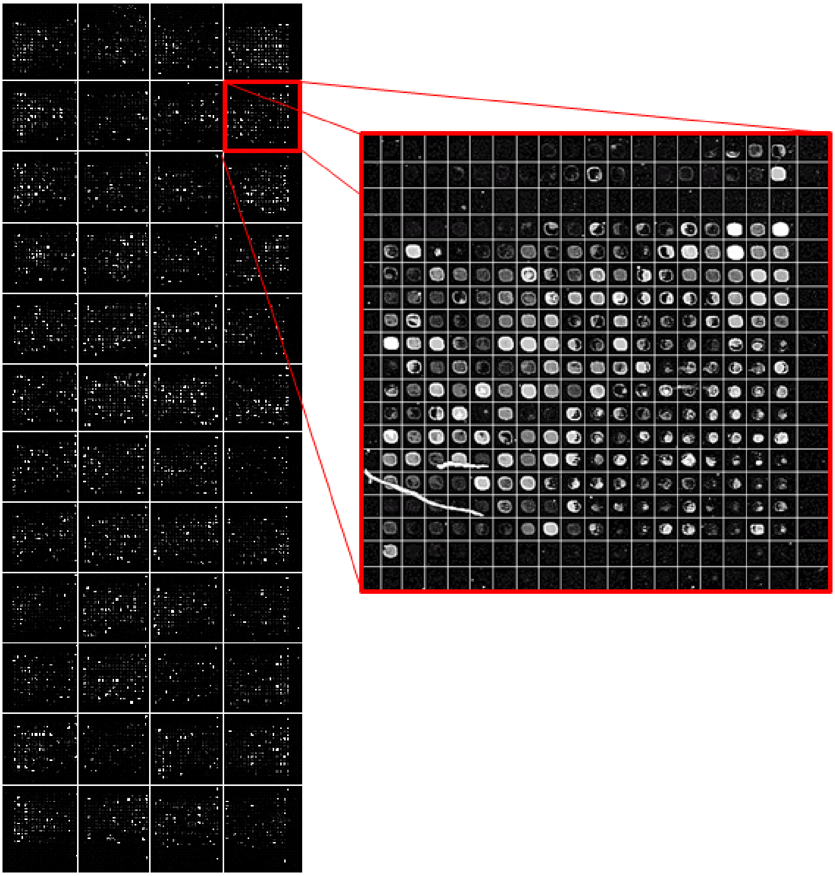


Fig. 3. Detected sub-grids in AT-20387-ch2 microarray image (left) and detected spots in one of sub-grids (right)

is divided nicely into the corresponding spots with the same method. The robustness of the proposed method is so high that spots in sub-grids can be detected very well in noisy conditions such as those observable in the selected sub-grid in Figure 3. The ability to detect sub-grids and spots in different microarray images with different resolutions and spacing (12 × 12 pixels for each spot in GEO dataset vs 24 × 24 pixels for each spot in SMD dataset) is another important feature of the proposed method.

To visually analyze the efficiency of the proposed method to automatically detect the correct number of spots and sub-grids, we show in Figure 4 a plot for the indices against the number of classes (sub-grids) for AT-20387-ch2. Sub-figures (a), (b) and (c) represent the values of the index functions for the horizontal lines for the I index, A index and β index respectively, while (d), (e) and (f) contain the plots of the indices for the vertical separating lines. We observe that it could be rather difficult to find the correct number of classes (sub-grids) using solo the I index or the A index,

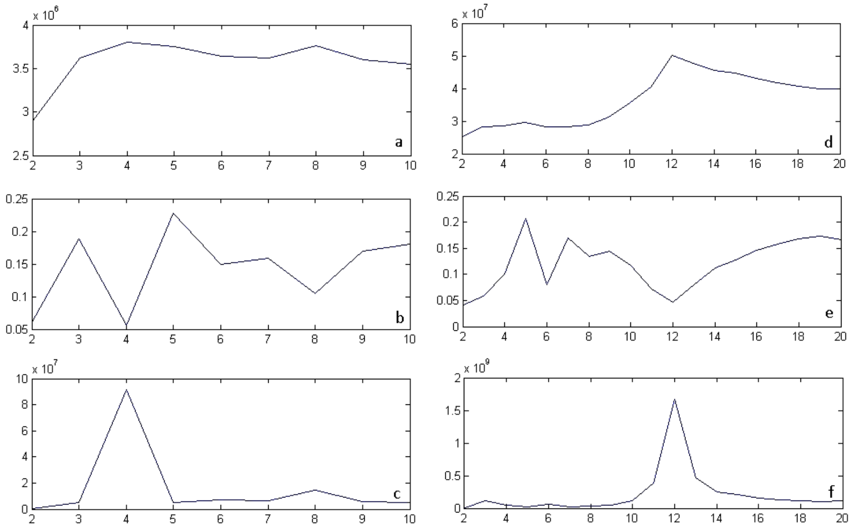


Fig. 4. Plots of the index functions for the AT-20387-ch2 microarray image. (a),(d): I index; (b),(e): A index; (c),(f): β index. The plots on the left correspond to the horizontal lines, while the ones on the right correspond to the vertical lines. the x axis corresponds to the number of classes (sub-grids).

while the β index clearly reveals the correct number of horizontal and vertical sub-grids by producing an almost flat curve with a pronounced peak at 4 and 12 respectively. For example, it is clearly observable in Figure 4 (a) that the I index could miss the correct number of sub-grids, 4, by showing two peaks (local and global maxima).

6 Conclusions

A novel method for separating sub-grids and spot centers in cDNA microarray images has been proposed. The method performs three main steps involving the Radon transform for detecting rotations with respect to the x and y axes, the use of polynomial-time optimal multilevel thresholding to find the positions of the lines separating sub-grids and spots, and a new index for detecting the correct number of sub-grids and spots. The proposed method has been tested on real-life, high-resolution microarray images drawn from two sources, the SMD and GEO. The results show that (i) the rotations are effectively detected and corrected by affine transformations, (ii) the sub-grids are accurately detected in all cases, even in abnormal conditions such as extremely noisy areas present in the images, (iii) the spots in each sub-grid are accurately detected using the same method, and (iv) because of using an optimal and parameterless algorithm for detecting threshold locations, this method can be used for microarray images with different features, and also for images with various spot sizes and configurations effectively.

References

1. Ceccarelli, B., Antoniol, G.: A Deformable Grid-matching Approach for Microarray Images. IEEE Transactions on Image Processing 15(10), 3178–3188 (2006)
2. Bariamis, D., Maroulis, D., Iakovidis, D.: M^3G: Maximum Margin Microarray Gridding. BMC Bioinformatics 11, 49 (2010)
3. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. John Wiley and Sons, Inc., New York (2000)
4. Zacharia, E., Maroulis, D.: Micoarray image gridding via an evolutionary algorithm. In: IEEE International Conference on Image Processing, pp. 1444–1447 (2008)
5. Qi, F., Luo, Y., Hu, D.: Recognition of perspectively distorted planar grids. Pattern Recognition Letters 27(14), 1725–1731 (2006)
6. Antoniol, G., Ceccarelli, M.: A Markov Random Field Approach to Microarray Image Gridding. In: Proc. of the 17th International Conference on Pattern Recognition, pp. 550–553 (2004)
7. Angulo, J., Serra, J.: Automatic Analysis of DNA Microarray Images Using Mathematical Morphology. Bioinformatics 19(5), 553–562 (2003)
8. Rueda, L.: Sub-grid Detection in DNA Microarray Images. In: Proceedings of the IEEE Pacific-RIM Symposium on Image and Video Technology, pp. 248–259 (2007)
9. Rueda, L.: An Efficient Algorithm for Optimal Multilevel Thresholding of Irregularly Sampled Histograms. In: Proceedings of the 7th International Workshop on Statistical Pattern Recognition, pp. 612–621 (2008)
10. Rueda, L., Vidyadharan, V.: A Hill-climbing Approach for Automatic Gridding of cDNA Microarray Images. IEEE Transactions on Computational Biology and Bioinformatics 3(1), 72–83 (2006)
11. Katzer, M., Kummer, F., Sagerer, G.: A Markov Random Field Model of Microarray Gridding. In: Proceeding of the 2003 ACM Symposium on Applied Computing, pp. 72–77 (2003)
12. Maulik, U., Bandyopadhyay, S.: Performance Evaluation of Some Clustering Algorithms and Validity Indices. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(12), 1650–1655 (2002)
13. Wang, Y., Ma, M., Zhang, K., Shih, F.: A Hierarchical Refinement Algorithm for Fully Automatic Gridding in Spotted DNA Microarray Image Processing. Information Sciences 177(4), 1123–1135 (2007)

Quantification of Cytoskeletal Protein Localization from High-Content Images

Shiwen Zhu[1], Paul Matsudaira[1,2], Roy Welsch[1,4], and Jagath C. Rajapakse[1,3,5]

[1] Computation and System Biology, Singapore-MIT Alliance, Nanyang Technological University, Singapore 637460
[2] Department of Biology Science, National University of Singapore, Singapore 117543
[3] School of Computer Engineering, Nanyang Technological University, Singapore 639798
[4] Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts 02142, USA
[5] Department of Biological Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA

Abstract. Cytoskeletal proteins function as dynamic and complex components in many aspects of cell physiology and the maintenance of cell structure. However, very little is known about the coordinated system of these proteins. The knowledge of subcellular localization of proteins is crucial for understanding how proteins function within a cell. We present a framework for quantification of cytoskeletal protein localization from high-content microscopic images. Analyses of high content images of cells transfected by cytoskeleton genes involve individual cell segmentation, intensity transformation of subcellular compartments, protein segmentation based on correlation coefficients, and colocalization quantification of proteins in subcellular components. By quantifying the abundance of proteins in different compartments, we generate colocalization profiles that give insights into the functions of different cytoskeletal proteins.

Keywords: Colocalization, cytoskeletal proteins, subcellular localization, cytoskeleton.

1 Introduction

The cytoskeleton is a cellular skeleton – a dynamic structure in all eukaryotic cells and some of prokaryotic cells – that function dynamically in many aspects including the maintenance of cell shape, the protection of cells, the organization of the cytoplasm, the support of the cellular machinery for motility, the transportation, the organization of cells into tissues and the signaling. Since cytoskeletal proteins are involved in so many functions, they are chemically connected to the reactions of metabolism and to the complex functional networks of small molecules and enzymes that transport signals within cells [1][2]. With those signals, cytoskeletal proteins generate harmonious responses to the coordinated efforts of cellular networks. However, very little is known about the coordinated system of these proteins. Investigation of the exact roles of cytoskeletal proteins, therefore, has become an

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 289–300, 2010.

important task that would greatly benefit many research areas including cellular mechanics, subcellular organization, metabolic signaling pathway modeling, early development of cancer, etc.

The knowledge of subcellular location of proteins is crucial for understanding how proteins function within a cell. Fluorescent microscopy has been used more and more frequently to identify protein subcellular locations through image processing, feature extraction, and pattern recognition [3][4]. Machine learning methods have been previously applied for identifying and predicting the localization patterns of proteins by using training data [5][6][7][8]. But such methods were intended for single cell and single channel data.

How different proteins interact with more than one subcellular compartment and their presence in more than one location has not been addressed. More cellular compartments need to be considered as predicting candidates for a protein's location and prediction of a single subcellular protein pattern is not sufficient. Thus we chose to focus on fluorescent signal colocalization – a measurement of overlap between two signals. Quantifying a single colocalization parameter is necessary so that a variety of proteins can be quickly and easily compared without bias. Determining the colocalization between cytoskeletal proteins and subcellular components—such as nucleus, cytoplasm, plasma membrane, actin network, etc., will help to define and simplify the proteins' operations and locations.

There are two basic ways to measure the colocalization [9]: global statistical approaches that perform intensity correlation coefficient based analyses; and object-based approaches. The global statistical approaches mainly use statistics to assess the relationship between fluorescence intensities in different compartments, including Pearson's coefficient [10], overlap coefficient [10], a statistical significance algorithm based on Pearson's coefficient [11], intensity correlation analysis [12], etc. However, the global statistical approaches rely on individual pixel coincidence analysis, globally providing colocalization estimation of the whole image but not of a unique structure. In the situation of low transfection efficiency, the cells with no or less GFP signals will pull down the overall colocalization estimates. Several methods of object-based approaches have been proposed such as comparing the position of the centroids or intensity centers of the objects [13] and normalized mean deviation product [14]. But they all focus on protein-protein colocalization analysis, which is the correlation between two different protein channels. In the protein subcellular colocalization analysis, the intensities within subcellular compartments are not a major concern. For example, DNA is stained to represent the nucleus. The DNA intensities in nuclear compartment may not be uniform, but our major concern is how the protein is colocalized in the nucleus. Therefore, we introduce a new colocalization measurement inducing colocalization profiles indicating the amount of colocalization of proteins and subcellular compartments.

Experiments were carried out on HeLa cell lines transfected with cytoskeletal protein genes. High content images of protein localization were measured and clolocalization profiles were generated. Statistical analysis showed that the cytoskeletal proteins can be clustered into several groups with similar colocalization patterns.

2 Method

In order to quantify subcellular localization of cytoskeletal proteins with a single parameter, the colocalization, we developed a computational framework involving individual cell segmentation, protein segmentation, intensity transformation of subcellular compartments, and colocalization computing. In what follows, we describe the different steps involved in our approach.

2.1 Cell Segmentation

Images were segmented into small objects using a multi-resolution segmentation technique based on object-oriented image analysis. This method is used to create object primitives as the first processing step in the segmentation analysis. The criterion for the segmentation is that average heterogeneity of image objects weighted by their size in pixels should be minimized. After the primary segmentation, the image objects are classified as nuclear objects and cell body objects based on flexible thresholds of nuclei and cell intensities. The nucleus objects are used as seeds for region growing method for cell segmentation, on the assumption that each cell has only one nucleus. A rule set was then developed for cell segmentation. The cell body object was fused with its neighbor nucleus. When one cell body object has more than one neighbor nuclear object, it was fused with the nuclear object that shared the largest border with it. The region growing method with multi-resolution segmentation provided better segmentation results than other advanced segmentation algorithms (Fig. 1): Level set method with shape marker and marking function [15] combined with nuclear information and level set method with topological dependence [16].

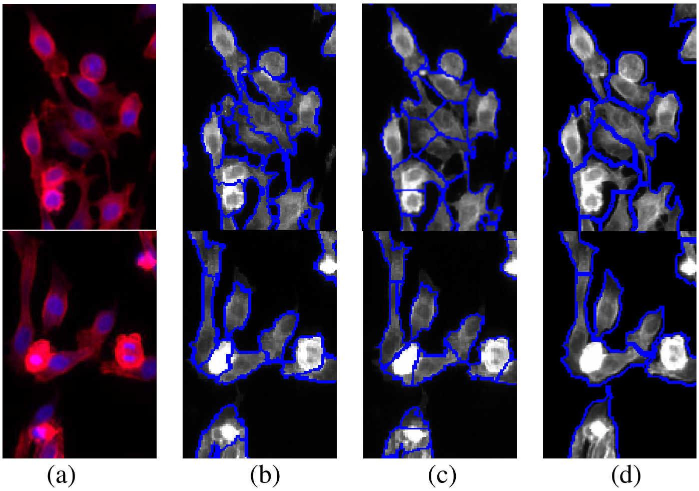


Fig. 1. (a) Images of actin and nuclear channels, and segmentation results of (b) region growing with multi-resolution segmentation, (c) adaptive level set method with shape marker combined with nuclear information, and (d) level set method with topological dependence

In the experiments, cells were labeled with fluorescence to highlight nuclei, actin, and proteins of the cells. The green fluoresce protein (GFP) was selected to identify a particular protein. The nuclear and actin channels were used to identify subcellular compartments. Let $f_n : \Omega \rightarrow R$, $f_a : \Omega \rightarrow R$ and $f_g : \Omega \rightarrow R$ represent the images of nuclear channel, the actin channel, and the GFP channel, respectively, where $\Omega \subset \mathrm{R}^2$ is the 2D image domain and $x \in \Omega$ denotes the 2-D coordinates of a pixel site in the image. Let $w_i \subset \Omega$ denote the area of an individual cell, and $i \in N$ be the labels of cells in one image.

2.2 Intensity Transformation of Subcellular Compartments

In this study, we are interested in five subcellular compartments: nucleus, cytoplasm, actin, plasma membrane, and cytosol. For most of the subcellular compartments, the intensity distributions of labeled signals were uniform, but the colocalization amounts of proteins were different. Thus the intensity transformations of subcellular compartments are performed using an intensity information of actin and nucleus staining as well as the position and relation information of subcellular compartments. Instead of computing the colocalization of the protein signal and the compartment signal, we compute the colocalization of the protein signal and the intensity transformed images of the compartments. Let a compartment be denoted by $c = \{nucleus, cytoplasm, actin, membrane, cytosol\}$, and $\tilde{f}_c : \Omega \rightarrow R$ denote the intensity transformed image in the compartment c .

Actin: This is the compartment identified from the actin channel. A ceiling threshold is used to account for over-saturation and the intensity values in the actin channel were rescaled to the range [0, 1].

$$\tilde{f}_{actin}(x) = \begin{cases} 1, \quad f_a(x) \geq t_a; \\ \dfrac{(f_a(x) - t_a')}{(t_a - t_a')}, \quad t_a > f_a(x) \geq t_a'; \\ 0, \quad f_a(x) < t_a'. \end{cases} \qquad x \in w_i; \tag{1}$$

where t_a and t_a' are positive numbers representing upper and lower thresholds of the actin channel.

Cytoplasm: This is the compartment identified as the non-nuclear region. The transformation for the cytoplasm compartment is kept uniform.

$$\tilde{f}_{cytosplasm}(x) = 1 - \tilde{f}_{nucleus}(x), \qquad x \in w_i; \tag{2}$$

Nucleus: This is the compartment identified from the nuclei channel. We keep the intensities of the nucleus compartment uniform:

$$\tilde{f}_{nucleus}(x) = \begin{cases} 1, & f_n(x) \geq t_n \\ 0, & f_n(x) < t_n \end{cases}, \quad x \in w_i; \tag{3}$$

t_n represents the threshold of the nuclei channel.

Plasma Membrane: This is the compartment identified as the border region of a cell. The intensities of plasma membrane were transformed using an exponential function of the minimum Euclidean distance to the cell border.

$$\tilde{f}_{membrane}(x) = t_m \cdot \exp\left[-\left(\frac{(d(x)-t'_m)}{t''_m}\right)^2\right], \quad x \in w_i; \tag{4}$$

Where $$d(x) = \min\sqrt{(x-x')^2}, \quad x \in w_i, x' \in \partial w_i; \tag{5}$$

and t_m, t'_m, t''_m are positive numbers representing the parameters of the exponential function with ∂w_i representing the border.

Cytosol: This is the compartment identified as cytoplasm without the components of actin and the plasma membrane.

$$\tilde{f}_{cytosol}(x) = \max\{0, \tilde{f}_{cytoplasm}(x) - \tilde{f}_{membrane}(x) - \tilde{f}_{actin}(x)\}, \quad x \in w_i; \tag{6}$$

2.3 Protein Segmentation

In order to identify the protein localization, cells were transfected with the protein tagged with GFP. By localizing the scattering of GFP-tagged proteins in the cells, its localizations in different subcellular compartments were identified. Before identifying the subcellular localization of the protein, the segmentation of protein needs to be correctly performed. Since the GFP intensities vary in different cells and compartments in different images, protein segmentation becomes a crucial component in finding the balance between capturing most of the protein information and highlighting the most specific protein information. Because the protein exists as small units, it cannot be segmented into one connected component. Thus, classical segmentation algorithms, such as watershed and region growing, become unsuitable. An automated thresholding method of identification of colocalized pixels has been earlier developed for protein-protein colocalization analysis [11].

Therefore, we develop an algorithm to segment the proteins by thresholding based on the correlation of its intensities with that of the responding compartment. The basic idea is to preserve most of the GFP pixels that are correlated with the intensities of the cell or its compartments, and remove the pixels that are least correlated or distributed almost randomly.

For each candidate threshold, the correlation coefficient is computed on both selected (intensities higher than the candidate threshold) and unselected pixels (intensities lower than the candidate threshold). Correlation coefficients high on selected pixels and low on unselected pixels indicate that the current thresholds can save high correlated pixels and remove low correlated pixels, respectively. Thus, we want to achieve the proper balance between these two thresholds.

Several correlation coefficients were tested such as Pearson correlation coefficient and overlap coefficient. The intensity correlation quotient (ICQ) provided the best segmentation results. For one given cell image with channel f_1, f_2, and a given image region w, the ICQ value is based on the intensity correlation coefficient ρ [17]. The correlation coefficient at a pixel is:

$$\rho(f_1, f_2, x) = (f_1(x) - \mu_1)(f_2(x) - \mu_2), \quad x \in w; \tag{7}$$

μ_1 and μ_2 denote the mean intensities values with the region w of the two channels.

The ICQ is defined as the ratio of the positively correlated pixels and the negatively correlated pixels in the region w. The correlation coefficient for two denoted areas is:

$$\rho(f_1, f_2, w) = \frac{\delta^+}{\delta^+ + \delta^-} - 0.5, \tag{8}$$

where $\delta^+ = \sum_{x \in w} \delta(\rho(x) > 0)$ donates the total number of positively correlated pixels, and $\delta^- = \sum_{x \in w} \delta(\rho(x) < 0)$ donates the total number of negatively correlated pixels. The range of ICQ falls between [-0.5 0.5]. When ICQ ≈ 0, random correlation; -0.5 ≤ ICQ < 0, negative correlation; 0 < ICQ ≤ 0.5, positive correlation.

The Algorithm 1 gives a way to determine the optimum threshold for protein segmentation within the cell. $f_{cell} : \Omega \rightarrow R$ is the cell image obtained by combining the nuclei channel and the actin channel: $f_{cell}(x) = f_a(x) + f_n(x), x \in w_i$. T is the final threshold generated for protein segmentation in this particular cell region w_i.

2.4 Colocalization

After segmentation of proteins within the cell and intensity transformation of subcellular compartments, the colocalization of proteins and subcellular compartments is quantified by a "colocalization" measurement that gives a better understanding about the percentage protein localized in a compartment. The colocalization of a protein p in a compartment c is defined as:

$$Coloc(c, p) = \frac{\sum_{x \in W_p} f_{\mathrm{p}}(x) \cdot \tilde{f}_c(x)}{\sum_{x \in W_p} f_{\mathrm{p}}(x)} \; ; \tag{9}$$

where W_p represents the set of pixels in the region occupied by the protein p . $f_{\mathrm{p}}(x)$ is the intensity distribution of the GFP channel highlighting protein *p* and $\tilde{f}_c(x)$ is the intensity transformation of the compartment c .

Algorithm 1. Determination of Threshold for Protein Segmentation

```
begin
  $t = \max_x \{f_g(x) : x \in w_i\}$
  $t_1 = t_2 = 0$
  $r_1 = 0$
  $r_2 = 1$
  while $t \geq 0$
      $W_{GFP} = \{x : f_g(x) \geq t, x \in w_i\}$
      $\bar{W}_{GFP} = \{x : f_g(x) < t, x \in w_i\}$
      if $\rho(f_{cell}, f_g, \bar{W}_{GFP}) < r_1$
      $t_1 = t$
      endif
      if $\rho(f_{cell}, f_g, W_{GFP}) > r_2$
      $t_2 = t$
      endif
      $t = t - 1$
  endwhile
  $T = \frac{t_1 + t_2}{2}$
end
```

2.5 Protein Localization Profiling

Using a library P of GFP-tagged cytoskeletal protein constructs, we compute the colocalization values of the protein in the five subcellular compartments. For given protein $p \in P$, the colocalization profile is $Col(p) = \{Coloc(c, p)\}$, where $c = \{nucleus, cytoplasm, actin, membrane, cytosol\}$, we then cluster those protein profiles in order to find the functional proteins.

3 Experiments and Results

3.1 Sample Preparation

Eighty-nine Invitrogen GFP-tagged cytoskeletal protein constructs were transfected to Hela cells in two 96-well plates (2 wells per construct, some constructs are duplicated). Each well has about 10,000 cells before transfection. Lipofectamine2000 transfections were taken to each well with constructs concentration 10ng/ul. Then the cells are fixed and stained with Hoechst33342 (nuclei) and Texas red phalloidin (Actin).

3.2 Imaging

Imaging of transfected cells was performed by Cellomics vHCS: Scan V Target Activation application system with 20X magnification. For each image sample there were 96 wells containing 48 constructs transfection results; and for each well there were 40 fields being scanned. Thus, the number of images in this dataset is about 7680. For each image, there are three fluorescent channels: blue (Hoechst33342) staining nuclei, red (Texas red - phalloidin) staining actin, and green (GFP) staining the particular protein.

3.3 Image Processing

After the high-content imaging, the images are analyzed with the computational frame work described in Methods section, involving individual cell segmentation, Intensity transformation of subcellular compartments, protein segmentation based on correlation coefficients, and colocalization quantification of proteins in subcellular components. For cell segmentation, a multi-resolution segmentation technique provided by Definiens Developer was used [18]. In the intensity transformation step, the fitting parameters for the exponential function are: t_m =1.054, t'_m =1.53, t''_m =2.241.

A colocalization matrix is generated with the dimension of 89 proteins × 5 subcellular compartments × the number of cells transfected with each protein. The colocalization matrix went through post data analysis to generate final conclusions.

3.4 Cell Selection

Before clustering the proteins, we apply a cell selection procedure based on nucleus area to delete part of the under-segmented cells as the nucleus area should have little variance in normal cells. The histogram of Nucleus Area feature is represented as a two peak curve, one peak is relatively smaller than the other. From biology we know that the higher peak shows the population of the normal nucleus and the smaller peak appears at the position where the area is twice of the normal nucleus area, showing the under-segmented two-connected nuclei. A single Gaussian-fit is applied to the histogram of nucleus area and the interval which contains 90% of the values from the fitting distribution was computed. The interval is [302,997]. Although the interval still contains some under-segmented nuclei, it successfully removes many of the miss-segmented cells.

In the current image dataset, the low transfection efficiency largely affects the final results of the colocalization analysis: a large number of cells are not transfected pulling down the total sample number; while the over-expressed ones show abnormal morphology and consequently abnormal colocalization results. After the transfection, the cytoskeletal proteins will first head to their normal subcellular locations or their functional locations. But in the over-expressed situation, more and more cytoskeletal proteins are generated and run everywhere inside the cell, which makes plenty of noise and reduces the significance of the functional locations of the particular cytoskeletal protein. Thus, we perform a GFP-intensity analysis to find the optimized intensity interval to eliminate abnormal-transfected cells.

In the GFP intensity analysis, the transfected cells are clustered into several GFP-intensity groups. By computing the colocalization values for each GFP intensity group, the colocalization trends along with the increasing GFP intensities are investigated and we concluded that the colocalization values do change greatly with the increasing GFP intensity. In the high GFP intensity intervals (greater than 50), we could find that the colocalization values in the Nucleus compartment increased, indicating that the over-expressed cells could round up, as dead cells or toxic cells, which been proved by observation. This phenomenon affects the overall colocalization results, especially the constructs with low transfection efficiency. To keep the particular colocalization pattern as well as to avoid noise, the intensity interval [20, 30] seems to be a good choice for the cell selection.

3.5 Colocalization Indexing

In order to provide a standard comparison between subcellular compartments, we perform k-means clustering separately on all 5 colocalization values. The sums of squared distances are examined to determine the best number of clusters. For each K, the K-means clustering is replicated 100 times to mitigate the effects of different initial conditions. Three is decided as the number of clusters as most of the colocalization values show inflection on it. The cluster labels from 1 to 3 are assigned to each protein to represent its colocalization degree for a specific subcellular compartment.

3.6 Protein Clustering

K-means clustering is performed again based on the cluster label of each protein for further protein classification. The sums of squared distances are examined to determine the best number of clusters. For each K, the K-means clustering is replicated 100 times to mitigate the effects from different initial conditions. Four is chosen as the number of the clusters and all the proteins are clustered into four clusters (Table 1): cluster 1 – 21 proteins with equally distributions within cells; cluster 2 – 21 proteins with high colocalization in plasma membrane; cluster 3 – 34 proteins with high colocalization in actin and cytosol; and cluster 4 – 13 proteins which are toxic to cells leading to cell round up. In Table 2 all proteins in each cluster are listed. As cytoskeletal proteins dynamically function within cells, the colocalization profiles will provide a distribution ratio among the subcellular compartments rather than one specific compartment. Although the exact functions of most of proteins remain unclear, the results can be validated with literature research. It is noticeable that some proteins in the same protein family are clustered together with similar colocalization profiles, such as TAGLN and TAGLN2, ITGB1and ITGB2, MYO3A and MYO1A, etc. Another validation is the comparison of reported functions of proteins and colocalization profiles. For example, cluster 3 shows high colocalization in actin compartment, and 22 of 34 proteins in this cluster are reported as having relative functions with the actin network.

Table 1. Colocalization of Protein Clusters

	Nucleus	*CytoP*	*CytoS*	*Actin*	*PM*	*protein No.*
Cluster 1	52.42% ± 4.54%	47.58% ± 4.54%	16.26% ± 3.31%	49.40% ± 5.41%	14.37% ± 5.81%	21
Cluster 2	56.63% ± 8.48%	43.37% ± 8.48%	7.85% ± 2.66%	43.96% ± 4.66%	37.60% ± 4.18%	21
Cluster 3	34.70% ± 6.08%	65.30% ± 6.08%	18.62% ± 3.36%	53.23% ± 7.53%	20.54% ± 5.50%	34
Cluster 4	75.72% ± 9.07%	24.28% ± 9.07%	6.81% ± 3.07%	45.99% ± 13.04%	15.57% ± 7.27%	13

Comparing the proteins within the same cluster and in different clusters, proteins with similar colocalization profiles are considered to have similar functions. For example, a set of proteins with unclear function such as filamin A interacting protein 1 (FILIP1) and tropomyosin 1 (TPM1), are seen to have similar profiles showing significantly high colocalization values in plasma membrane (cluster 2) together with other proteins having related functions with plasma membrane such as integrin beta 1 (ITGB1), integrin beta 2 (ITGB2), and villin 2 (VIL2).

Table 2. Proteins in Protein Clusters

	protein No.	*Protein Brief Name*
Cluster1	21	CORO2B,PDLIM3,DNM2,TAGLN,TNS,TEKT3,PTK9,PLS1,JAMIP2, ATP1B3,VAMP4,PXN,MSN,ADRM1,MRLC2,TAGLN2,ARPC5,VIM, NINJ2,PFN2,PARVA
Cluster2	21	TUBA6,TGOLN2,CORO1B,ATP6V1C1,TAGLN3,VIL2,TUBD1, TUBGCP3,PARVG,RDX,CAPZA3,ACTG1,PTK9,TPM1,ITGB1,ITGB2, ADAM15,ARHGEF6,EIF2C1,WASF2,FILIP1
Cluster3	34	DCAMKL1,WASPIP,KLHDC9,CTTN,VIL1,ACTB,WASL,ZYX,CDC42, CFL1,EVL,TUBE1,CLIP3,DSTN,PAK4,VASP,LPXN,KIF2C,ITGB7, PLS3,ARPC1B,TUBG1,GTSE1,ACTN2,MYO3A,ACTN1,ARP11,CNN3, LCP1,TPM2,MYO1A,ANLN,FSCN1,KRT8
Cluster4	13	WAS,ATP5G2,ITGB3BP,FSCN3,GJB2,TUBA1B,KPTN,TEKT1,TUBB2, KAT5,WASF3,DCTN1,CAV3

4 Conclusions

In this work, we developed a computational framework and optimized every step in the framework to quantify the subcellular localization of cytoskeletal proteins with a single colocalization measurement. The framework is applied on a two-dimensional image set, containing around 8000 images of cells transfected with 89 cytoskeletal protein constructs. The subcellular localizations of those cytoskeletal proteins are quantified and localization patterns are investigated to provide references in investigation of protein functions. Proteins with unknown functions can be investigated by comparing with colocalization profiles generated in a cytoskeletal protein library.

For image-based subcellular localization quantification, two-dimensional analysis is not sufficient. In future work, the whole framework will be transferred to the three-dimensional domain. The quantification of subcellular localization on three-dimensional space will provide more accurate results. The quantification of subcellular localization will greatly benefit the investigation of functions of cytoskeletal proteins.

Acknowledgments. This work was partly supported by Computation and Systems Biology programme of the Singapore-MIT Alliance.

References

1. Khurana, S., Bittar, E.: Aspects of the cytoskeleton. Elsevier Press, Amsterdam (2006)
2. Bray, D.: Cell Movements: From Molecules to Motility, 2nd edn. Taylor & Francis Press, Abington (2001)
3. Hu, Y., Murphy, R.F.: Automated interpretation of subcellular patterns from immunofluorescence microscopy. Immunol Methods 290, 93–105 (2004)
4. Huang, M.R.F.: Boosting accuracy of automated classification of fluorescence microscope images for location proteomics. BMC Bioinformatics 5, 78–96 (2004)
5. Boland, M.V., Murphy, R.F.: A Neural Network Classifier Capable of Recognizing the Patterns of all Major Subcellular Structures in Fluorescence Microscope Images of HeLa Cells. Bioinformatics 17, 1213–1223 (2001)
6. Chen, X., Murphy, R.F.: Objective clustering of proteins based on subcellular location patterns. Journal of Biomedicine and Biotechnology, 87–95 (2005)
7. Hamilton, N., Pantelic, R., Hanson, K., Teasdale, R.: Fast automated cell phenotype image classification. BMC Bioinformatics 8, 110 (2007)
8. Chen, S.-C., Zhao, T., Gordon, G.J., Murphy, R.F.: Automated Image Analysis of Protein Localization in Budding Yeast. Bioinformatics 23, i66–i71 (2007)
9. Bolte, S., Cordelieres, F.P.: A guided tour into subcellular colocalization analysis in light microscopy. Journal of Microscopy 224(3), 213–232 (2006)
10. Manders, E.M., Stap, J., Brakenhoff, G.J., Driel, R., Aten, J.A.: Dynamics of three-dimensional replication patterns during the S-phase, analysed by double labelling of DNA and confocal microscopy. J. Cell Sci. 103, 857–862 (1992)
11. Costes, S.V., Daelemans, D., Cho, E.H., Dobbin, Z., Pavlakis, G., Lockett, S.: Automatic and quantitative measurement of protein-protein colocalization in live cells. Biophys. J. 86, 3993–4003 (2004)

12. Li, Q., Lau, A., Morris, T.J., Guo, L., Fordyce, C.B., Stanley, E.F.: A Syntaxin 1, Galphao, and N-type calcium channel complex at a presynaptic nerve terminal: analysis by quantitative immunocolocalization. J. Neurosci. 24, 4070–4081 (2004)
13. Boutte, Y., Crosnier, M.T., Carraro, N., Traas, J., Jeunemaitre, B.S.: Immuno cytochemistry of the plasma membrane recycling pathway and cell polarity in plants: studies on PIN proteins. J. Cell Sci. 113, 1255–1265 (2006)
14. Jaskolskia, F., Mullea, C., Manzonib, O.J.: An automated method to quantify and visualize colocalized fluorescent signals. J. Neurosci. Meth. 146, 42–49 (2005)
15. Cheng, J.R., Rajapakse, J.C.: Segmentation of clustered Nuclei with Shape Markers and Marking Function. IEEE Transactions on Biomedical Engineering 56(3), 741–748 (2009)
16. Yu, W., Lee, H.K., Hariharan, S., Bu, W., Ahmed, S.: Quantitative neurite outgrowth measurement based on image segmentation with topological dependence. Cytometry Part A 75A, 289–297 (2009)
17. McMaster Biophotonics Facility, `http://www.macbiophotonics.ca/PDF/MBF_colocalisation.pdf`
18. Baatz, M., Schäpe, A.: Multiresolution Segmentation –an optimization approach for high quality multi-scale image segmentation. In: Strobl, J., Blaschke, T., Griesebner, G. (eds.) Angewandte Geographische Informationsverarbeitung XII, pp. 12–23. Wichmann, Heidelberg (2000)

Pattern Recognition for High Throughput Zebrafish Imaging Using Genetic Algorithm Optimization

Alexander E. Nezhinsky and Fons J. Verbeek

Imaging and Bioinformatics, Leiden University, Leiden Institute of Advanced Computer Science, Niels Bohrweg 1, 2333CA, Leiden, The Netherlands

Abstract. In this paper we present a novel approach for image based high–throughput analysis of zebrafish embryos. Zebrafish embryos can be made available in high numbers; specifically in groups that have been exposed to different treatments. Preferably, the embryos are processed in batches. However, this complicates an automated processing as individual embryos need to be recognized. We present an approach in which the individual embryos are recognized and counted in an image with multiple instances and in multiple orientations. The recognition results in a mask that is used in the analysis of the images; multichannel images with bright–field and fluorescence are used.

The pattern recognition is based on a genetic algorithm which is the base of an optimization procedure through which the pattern is found. The optimization is accomplished by a deformable template that is incorporated in the genetic algorithm. We show that this approach is very robust and produces result fast so that it becomes very useful in a high–throughput environment. The method is fully automated and does not require any human intervention. We have tested our approach on both synthetic and real life images (zebrafish embryos). The results indicate that the method can be applied to a broad range of pattern recognition problems that require a high–throughput approach.

1 Introduction

Retrieving location and contour of a shape is crucial to the analysis of large image datasets in many fields, such as optical character recognition (OCR) and bio–imaging. If the number of occurrences of an object under study is not known beforehand, or the object we are trying to locate is subject to slight deformations extra complications arise; this is typical for life–sciences. In addition we sometimes need to take into account partial occlusion and noise.

In the current practices segmentation is a first step to separate objects from the background and a variety of segmentation techniques is available [8], [15]. In this paper we want to address the problem of recognition of object localization under different conditions of strain stress. Our specific interest is in shapes of which a prior shape information is available. To that end we need to use an approach that depends on the inexact predefined shape and can be subject to

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 301–312, 2010.

deformations in the input image. A deformable template [1] approach gives us the possibility to represent an object that can be subject to certain deformations in a compact manner. In that case object localization should be performed by a process of matching the deformable template to the object shape in the input image.

One approach is the *free–form* class [1] with the Active Contour (a.k.a. active snake) model [9]. This method has no global template structure and needs a starting location in the image to evolve from. Shape edges need to be connected together for an active snake to perform correctly. If the starting location is not known or multiple objects are present in the image this algorithm might be hampered.

We consider the type of deformable template models of the *parametric* class ([1]); i.e. the template shape is predefined as a set of parameters. In most approaches [15], [3], [11] deformable template representation of a shape is a set of points approximating the outline as obtained from a priori knowledge. In this manner a user defined input template is represented and we adopted this representation for our approach.

Matching a deformable template to an image can be seen as optimization problem with some possible global maxima – in our case best solution, being the best shape. This process is computationally expensive. A possible solution is therefore to reduce the search space [16] by focusing only on areas containing certain intensity (color). However, we want to base our algorithm on shape characteristics only, since color information might not always be available or subject to large variation. Genetic Algorithms (GA's) [4], [14] are typically suitable for solving global optimization problems, in particular if the solution space is very large. Therefore, we consider a Genetic Algorithm for optimization.

In this paper we introduce a slice representation model (SR) of a deformable template. Instead of considering the outline of a shape, we simplify the shape representation by considering only certain characteristic horizontal slices and use these for template matching. The proposed SR model is made advantageous for efficient optimization with a Genetic Algorithm. [6], [12] also used a GA for low parameter shape templates (circle and ellipse detection).

In the approach discussed in this paper we will use binary images as input. Binary images are successfully used for template and polygon matching in [10] and [7]). Whenever the images are presented as RGB or gray scale, they are thresholded to binary, cf [13].

In addition we want to address an important feature of automatic retrieval of multiple deformed shapes from a single image and counting the amount of shapes.

The paper is organized as follows. In section 2 a more detailed overview of our deformable template approach is given. Section 3 addresses the proposed GA used for shape recognition. In Section 4 we propose the application of the algorithm to retrieve multiple shapes in a single image. In section 5 experimental results for an application to a real life problem are shown and they are discussed in section 6.

2 Deformable Template

We propose a template representation which captures both boundary and interior of the object and thereby describes the average structure of a predefined shape. The grid of the search space is determined by the discrete pixels in a $K \times L$ image matrix M. We take a binary image as starting point. The binary image is obtained by thresholding. Thereby we assume a background pixel has the value 0 and a foreground pixel value 1. In the following subsections we will describe the template representation in more detail as well as the deformations that a template can undergo.

2.1 Slice Representation Model

The prototype template T_0 we propose is represented by the following vector:

$$T_0 = (s_0, s_1, .., s_n, d), \tag{1}$$

where n is the number of slices; d is the distance between two slices in the horizontal direction. Initially T_0 is represented in a X(horizontal)–Y(vertical) space in the horizontal direction, with slices being parallel to the Y–axis. This is done for ease in the representation. A template slice itself is then represented by the vector:

$$s_i = (w_i, b_i, \varsigma_i), \tag{2}$$

where w_i is the fixed width in pixels in the vertical direction, preferably $w_i = 1$; where b_i is the fixed width in pixels in the vertical direction of the image, preferably $b_i = 0$ and are located below and above the area of w_i. The b_i are required to define that the area surrounding the shape is preferably empty. ς_i is the seed point of a slice; ς is needed to be able to define non symmetrical shapes as input. This point will be used as a reference for allowed slice shifts. In Fig. 1(a) an example slice is shown; a slice is a sample of the template always perpendicular to the length axis.

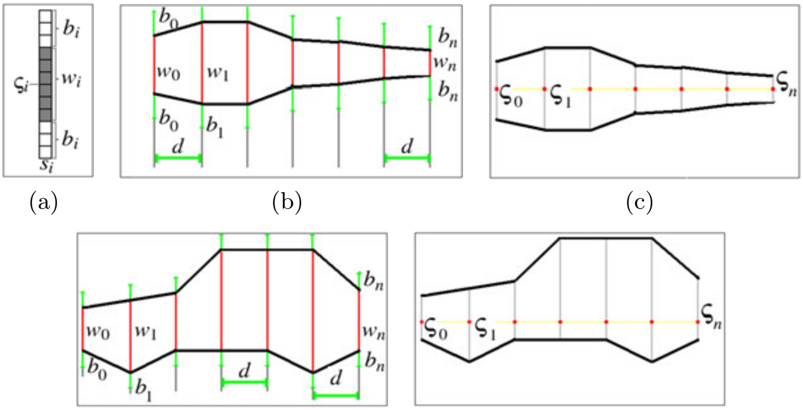
In Fig. 1(b), 1(c), 1(d) two examples of templates are shown. The template length is fixed according to $\ell(v) = n * d$. For templates with horizontal symmetry axis, ς is simply chosen as the center of each slice (cf Fig. 1(c)). For templates having no horizontal symmetry axis, ς is chosen in such a way that all ς_i are located on the same horizontal line (cf Fig. 1(e)).

2.2 Deformations

This representation was chosen as we assumed the slices will be able to move vertically along the template to match a deformed (slightly shifted) shape. At template shift or slight rotation the global shape is still captured within the template. The total length of the template is fixed in the proposed representation.

A deformed template T is derived from the prototype template T_0 and is represented as $T(T_0, I)$. A deformation I will then be encoded by a state-sequence as follows:

$$I = (x, y + \delta_0, y + \delta_1, .., y + \delta_n), \tag{3}$$



(a) (b) (c)

(d) (e)

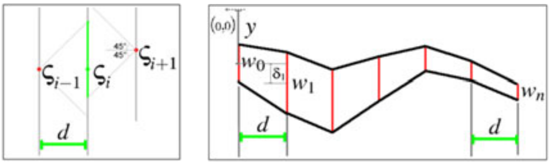
Fig. 1. a) A single slice. The slice is always parallel to the Y–axis b) Template for a fish shape c) Location of ς in a fish shape template d) Template for a car shape e) Location of ς in a car shape template.

where x is the shift in the X-axis direction of ς_0; $y+\delta_i$ is the translation measured in the Y–axis direction of the ς_i of slice i.

We assume the maximal vertical deformation between two slices is 45 degrees (cf Fig. 2(a). As a result of this constraint $|\delta_{i-1} - d| < |\delta_i| < |\delta_{i-1} + d|$ applies for two starting points of consecutive slices i and $i + 1$. Then, each ς_i can then be shifted vertically within the following boundaries:

$$\max\{(\varsigma_{i-1} - d, \varsigma_{i+1} - d)\} < \varsigma_i < \min\{(\varsigma_{i-1} + d, \varsigma_{i+1} + d)\}. \tag{4}$$

In Fig. 2(b) we demonstrate a possible deformation of a template representing a fish.



(a) (b)

Fig. 2. a)Vertical shift margins allowed for slice starting point ς_i, relative to ς_{i-1} and ς_{i+1}. b)A possible deformation for a fish template.

2.3 Energy Function

The energy function Φ, is a fitness function that specifies to what extent an identified shape in the image matches the deformed template. In the literature the probability of a deformed template being located over a shape is referred to as the likelihood [1].

Let us now introduce the details of the computation of Φ:

Consider the fitness ϕ_i of a single slice i. In order to find the optimum we wish to maximize matching values (0 or 1) between pixels covered by the slice, i.e. pixels with value 1 contained in w_i and pixels (at top and bottom of the slice) with value 0 in b_i. $S_i[j]$ represents the j–th element (pixel) of slice S_i, as counted from the top of the slice.

$$\phi_i = \frac{1}{w_i + b_i * 2} \sum_{j=0}^{w_i+2*b_i} H_j, H_j = \begin{cases} 1, & \text{if } M(x, j + y + \delta_i) = S_i[j] \\ 0, & \text{otherwise} \end{cases}$$

In Fig. 3 an example of a matching is given. The location of proposed matching is denoted in pattern. For this example:

$$\phi_i = \frac{1+0+0+1+1+1+1+1+0+1+1+1}{12} = 0.75$$

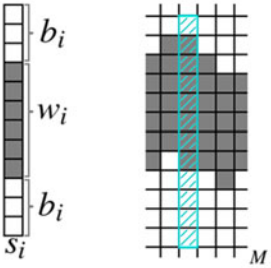


Fig. 3. Example of matching a slice S_i on to a location in matrix M

The deformation I consists of multiple slice shifts. Total fitness of all n slices is given by:

$$\Phi = \frac{1}{n} \sum_{i=0}^{n} \phi_i. \tag{5}$$

3 Genetic Algorithm

In this section we discuss the major components of a Genetic Algorithm (GA), i.e., population, evaluation, selection, crossover and mutation respectively. In a GA, a population P (of size m) of candidate solutions is evolved toward better solutions by introducing computer analogues for recombination, mutation and selection.

A candidate solution is also referred to as an *individual.* The outline of a generic GA pseudo code reads:

```
$t = 0$
Initialize $P(t)$
Evaluate $P(t)$
while not terminate do
    $P'(t)$ = SelectMates $(P(t))$
    $P''(t)$ = Crossover $(P'(t))$
    $P'''(t)$ =Mutate $(P''(t))$
    Evaluate $P'''(t)$
    $P(t+1) = P'''(t)$
    $t = t + 1$
end while
```

The termination criterion can differ, depending on the problem at hand. The details of the operators are dealt with in more detail in the following subsections.

3.1 Representation of Individuals

A shape I based on the template T_0, also referred to as *individual* is then represented in the image as, and consists of the following genomes:

$$I = (x, y + \delta_0, y + \delta_1, .., y + \delta_n). \tag{6}$$

The individuals are initialized with randomly valued genomes. According to common practice in the GA research, a population is generated with random genome values, covering the entire range of possible solutions. For finding shapes in images, this means random shapes are initialized on random location in the target image with a random deformation according to Eq. 4.

3.2 Evaluation

The fitness function determines the quality of an individual and depends on the problem at hand. Function Φ (cf. Eq.5) will serve as a fitness function for the genome values of an individual; Φ is then used to evaluate the candidate solutions in the selection step.

3.3 Selection

For the selection operation the, so called, tournament selection scheme [2] is used in order to prevent premature convergence. Tournament size, i.e. k_{size}, was determined through empirical testing. Consequently, comparison of high and low k_{size} is given in Fig. 4. We have established $k_{size} = 7$ to be a good value for our experiments.

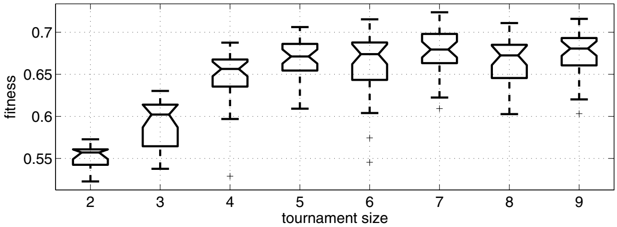


Fig. 4. Fitness comparison for different threshold values in 30 runs

3.4 Crossover

A crossover is applied with single crossover point. A random point $p \in (0, n)$ is chosen where n denotes the length of the genome. After crossover of two individuals I_J and I_K the resulting individual I_L has the following form:

$$I_L = (x_K, y_K, \delta_{0_K}, .., \delta_{p_K}, \delta_{p+1_J} + a, .., \delta_{n_J} + a). \tag{7}$$

Variable a is needed to make sure Eq. 4 holds and is determined by:

$$a = \begin{cases} \delta_{p_K} - d, & \text{if}(\delta_{p_K} - \delta_{p+1_J} > d) \\ \delta_{p_K} + d, & \text{if}(\delta_{p_K} - \delta_{p+1_J} < -d) \\ 0, & \text{otherwise} \end{cases}$$

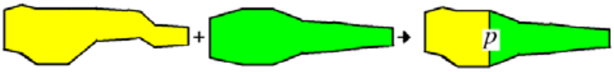


Fig. 5. Graphical representation of crossover function as derived from the data. Typical example in zebrafish imaging.

3.5 Mutation

For the mutation operator we use standard settings commonly used for GA's. That is a uniform mutation, where each genome g has the probability to mutate: $1/n$ [14], where n is the genome length. For each genome:

$$g \; U(\bar{g}, \underline{g})$$

To make sure a uniform mutation is used we allow every slice center to mutate anywhere within the image space. Since the constrain $\max\{(\varsigma_{i-1} - d, \varsigma_{i+1} - d)\} < \varsigma_i < \min\{(\varsigma_{i-1} + d, \varsigma_{i+1} + d)\}$ holds, subsequent slice centers are not allowed to be more separated then distance d.

4 Multiple Object Recognition

The shape under study can have multiple slightly different (deformed) instances in one and the same image. The complete search space in this image is denoted as M_0. First the best matching shape S_0 (with highest fitness) is localized. To decrease the search space for finding the next shape instance, we set all the elements contained within $S_0 \in M_0$ to 0 and name the resulting image M_1.

This process is repeated by iteration and in that manner M_i is reduced for each next identified shape i. No shapes are found if fitness of the found optimum $F(S_i)$ drops drastically; under a predefined threshold value r. The value of r can be determined empirically from a test on similar images (acquired under the same conditions). In Figure 6 an example of such drastic fitness drop in fitness growth is depicted. This is a fitness plot for an image containing 3 fish shapes (cf Fig. 8(a)).

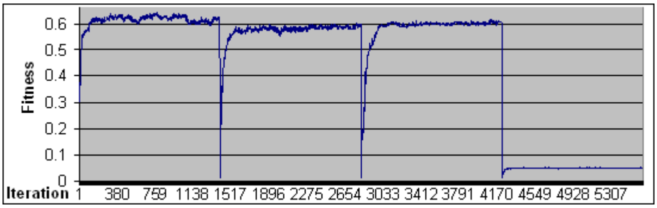


Fig. 6. Fitness evolution of 3 fish shapes found in an image. Every 1500 generations the found shape S_i is extracted and the algorithm is restarted on M_{i+1}. After finding S_0, S_1 and S_2 with fitnesses over 0.5, at the fourth run the maximum fitness can not get over 0.1. This is what we consider a fitness drop. For this type of images r should be chosen somewhere between 0.1 and 0.5, e.g. 0.4 is a good value.

The pseudo code for finding multiple shapes in an image can be written as:

```
$i = 0$
$S_0 = \mathrm{GA}(M_0)$
Evaluate $P(t)$
while $F(S_i) > r$ do
    save $S_i$ as found shape
    $M_i = M_{i-1} - S_i$
    $i = i + 1$
    $S_i = \mathrm{GA}(M_i)$
    $P(t+1) = P'''(t)$
    $t = t + 1$
end while
```

5 Experiments

To evaluate the performance of our template representation and GA optimization we have designed the task of finding objects based on predefined slice templates in images with different type of content. An experiment was performed with templates i.e. in synthetic as well as microscope images. With a simple interface the user selects both input images as well as the template shape for analysis.

5.1 Testing with Synthetic Images

We have used 20 synthetic binary images of 388 × 291 pixels. We have generated the images with different shapes located at different locations in images, slightly rotated, skewed and missing pixel data. Random noise (drawing debris) is introduced. The images randomly contain up to 3 instances of an object.

First template used for the synthetic shapes is a template of an animal figure presented in Fig. 7(a). We have created a synthetic binary image containing one deformed animal shape. The result of the application of the GA optimization is depicted in Fig. 7(c).

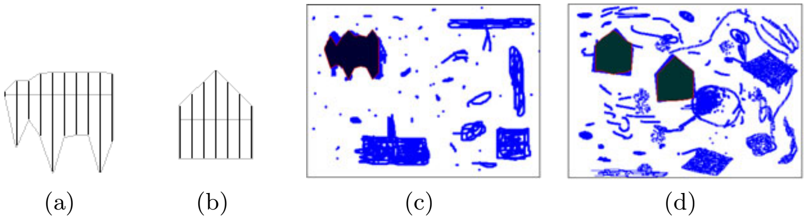


Fig. 7. a)Very simple template of an animal shape (head, legs and body) b) Very simple template of an house shape c) Result of shape localization in a synthetic image containing a simple animal shape d) Result of shape localization in a synthetic image containing two simple house shapes

Second template used for the synthetic shapes is a template of a house figure presented in Fig. 7(b). We have created a synthetic binary image containing two figures that have a deformed house shape. The result of the algorithm (implementation done in Delphi) is shown in Fig. 7(d).

5.2 Testing with Zebrafish Images

To evaluate the performance of our algorithm in a real–world imaging application we have chosen the task of finding zebrafish embryo shapes. The task at hand concerned using a High Throughput (HT) segmentation technique for retrieving the location and number of zebrafish embryo objects within images [5]. An additional requirement for this application is the need for automatic recognition of head, body and tail of each embryo. A typical binary image as presented for localization is shown in Fig. 8(a). This image was converted from color scale images to binary in a preprocessing step. We use a straightforward gradient operator (Sobel) followed by an iso–data threshold. In that way strong edge pixels were extracted for a binary representation [13].

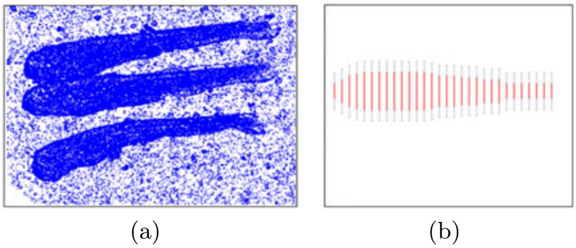

(a) (b)

Fig. 8. a) Typical binary image of zebrafish embryos in a resolution of 388 × 291 pixels. b) The template T_0 in a graphical representation that has been used on 100 tested images. Red lines represent w_i and gray represent b_i within slices. Some results are shown in Fig. 9.

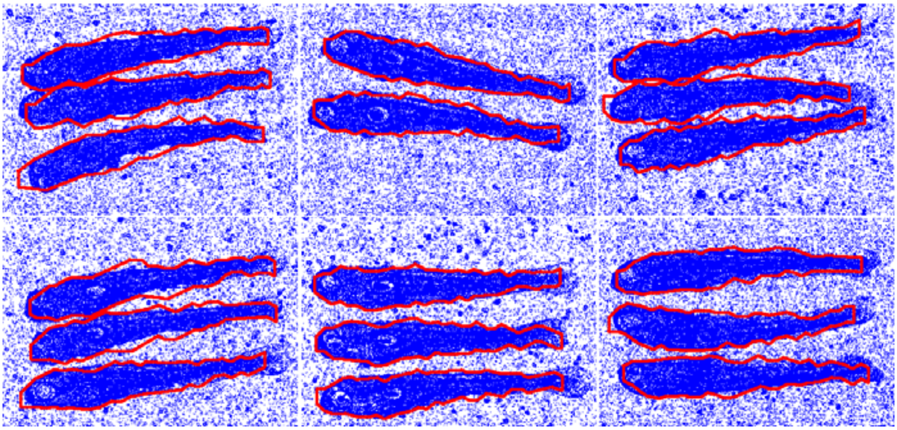


Fig. 9. Some results of shape counting and localization by our algorithm (Delphi implementation) on binary images. Rotated, slightly overlapping and bended objects could be retrieved. The algorithm needed about $2s$ CPU time for the retrieval of one shape on a Intel Dual Core 2.66Ghz, 1.00Gb.

Each image contains multiple zebrafish embryo shapes. The number of shapes in an image is not known in advance, the shapes might be overlapping. The shapes in all the images are assumed to be located approximately horizontal with a maximum angle of 45 degrees; so our approach could be used. We have applied the algorithm for a database of 100 images with the same settings for the GA ($m = 200$, $k_{size} = 7$, $r = 0.5$). For 87 images the amount of embryos in the image and the approximation of their shape could be retrieved correctly. In the cases where the algorithm failed, it was mostly due to large occlusion overlap or shapes were much longer or shorter than the proposed template. For the small and medium occlusions and overlap the algorithm performed correctly (cf Fig. 9). In Fig. 8(b) the template of a zebrafish used in these results is shown in a graphical representation.

6 Conclusions and Discussion

In this paper we have illustrated an application of a GA to optimize a deformable template approach. This method can be used in different fields as the template can represent different shapes. Our approach was designed for an application in the HT screening of zebrafish embryos [5]. The optimization of template parameters is done through a Genetic Algorithm, which provides the possibility to search for an optimal solution in large search spaces. Our approach also allows to retrieve multiple instances of a certain object in a single image. Results indicate that this approach has a low error rate while computational performance is manageable and fast, such is, of course, very suitable for HT applications. Future work is directed towards a further generalization of the approach and making the template representation scalable.

Acknowledgments

This work was supported by the Smartmix program. Special thanks for providing the images to Drs. E. Stoop and Dr. A. vd Sar and for the suggestions to Dr. M. Emmerich and Drs. J.Kruisselbrink.

References

1. Jain, A.K., Zhong, Y., Lakshmanan, S.: Object matching using deformable templates. IEEE Tran. on Pattern Analysis and Machine Intell. 18(3) (1996)
2. Miller, B.L., Goldberg, D.E.: Genetic algorithms, tournament selection, and the effects of noise. Complex Systems (1995)
3. Kervrann, C., Heitz, F.: A hieraerchial statical framework for the segmentation of deformable objects in image sequences. IEEE Comput. Vision Pattern Recogn. (1994)
4. Goldberg, D.E.: Genetic algorithms in search, optimization and machine learning. Kluwer Academic Publishers, Dordrecht (1989)
5. Stoop, E., et al.: Zebrafish embryo screen for mycobacterial genes involved in granuloma formation reveals a novel esx-1 component (submitted) (2010)
6. Yao, J., et al.: Fast robust ga-based ellipse detection. ICPR 2(2) (2004)
7. Krolupper, F., Flusser, J.: Polygonal shape detection for recogn. of partially occluded objects. Pattern Recogn. Lett. 28, 1002–1011 (2007)
8. Shapiro, L., Stockman, G.: Comput. Vision. Prentice-Hall, Englewood Cliffs (2002)
9. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. Journal of Comput. Vision. 1(4) (1987)
10. Funobiki, N., Isogai, M.: An eye-contour extraction algorithm from face image using deformable template matching. Mem. of the Fac. of Eng. 40, 78–87 (2006)
11. Nohre, R.: Deformed template matching by the viterbi algorithm. Pattern Recogn. Lett. 17(14) (1996)
12. Ramirez, A.: Circle detection on images using genetic algorithms. Pattern Recogn. Lett. 27(6) (2006)

13. Gonzales, R., Woods, R.: Digital Image Process, 2nd edn. Addison-Wesley, London (2001)
14. Bäck, T.: Evol. Algorithms in Theory and Practice: Evol. strategies, Evol. Programming, Genetic Algorithms. Oxford Univ. Press, Oxford (1996)
15. Cootes, T.F., Taylor, C.J., Cooper, D.H., Graham, J.: Active shape models - their training and application. Comput. Vision and Image Understanding 61(1) (2009)
16. Zhong, Y., Jain, A.K.: Object localization using color, texture and shape. Pattern Recogn. 33 (2000)

Consensus of Ambiguity: Theory and Application of Active Learning for Biomedical Image Analysis

Scott Doyle and Anant Madabhushi*

Department of Biomedical Engineering,
Rutgers University, USA
scottdo@eden.rutgers.edu, anantm@rci.rutgers.edu

Abstract. Supervised classifiers require manually labeled training samples to classify unlabeled objects. Active Learning (AL) can be used to selectively label only "ambiguous" samples, ensuring that each labeled sample is maximally informative. This is invaluable in applications where manual labeling is expensive, as in medical images where annotation of specific pathologies or anatomical structures is usually only possible by an expert physician. Existing AL methods use a single definition of ambiguity, but there can be significant variation among individual methods. In this paper we present a consensus of ambiguity (CoA) approach to AL, where only samples which are consistently labeled as ambiguous across multiple AL schemes are selected for annotation. CoA-based AL uses fewer samples than Random Learning (RL) while exploiting the variance between individual AL schemes to efficiently label training sets for classifier training. We use a consensus ratio to determine the variance between AL methods, and the CoA approach is used to train classifiers for three different medical image datasets: 100 prostate histopathology images, 18 prostate DCE-MRI patient studies, and 9,000 breast histopathology regions of interest from 2 patients. We use a Probabilistic Boosting Tree (PBT) to classify each dataset as either cancer or non-cancer (prostate), or high or low grade cancer (breast). Trained is done using CoA-based AL, and is evaluated in terms of accuracy and area under the receiver operating characteristic curve (AUC). CoA training yielded between 0.01-0.05% greater performance than RL for the same training set size; approximately 5-10 more samples were required for RL to match the performance of CoA, suggesting that CoA is a more efficient training strategy.

* Funding provided by the Wallace H. Coulter Foundation, New Jersey Commission on Cancer Research, National Cancer Institute (R01CA136535-01, R21CA127186-01, R03CA128081-01), Cancer Institute of New Jersey, US Dept. of Defense (W81XWH-08-1-0145), Bioimagene, Inc.

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 313–324, 2010.

1 Introduction

1.1 Using Consensus Methods for Certainty and Ambiguity

Ensemble classification algorithms such as bagging, boosting [1], and random forests [2] rely on some concept of consensus among several "weak" classifiers to generate a single "strong" result. Consensus, in the context of ensemble learning, describes agreement among several classification algorithms. For example, given a data object $\mathbf{x} \in \mathbb{R}^N$ belonging to one of c classes, $\omega_1, \cdots, \omega_c$, we can construct L classifiers $\mathcal{C}_l(\mathbf{x})$, for $l \in \{1, 2, \cdots, L\}$. The probability that $\mathbf{x}$ belongs to class ω_j, for $j \in \{1, 2, \cdots, c\}$, according to classifier l is denoted $p_l(\omega_j|\mathbf{x})$. While several classifier ensemble strategies seek to combine the weak learners using different rules, the underlying spirit of these methods is to assign the sample to the class ω_j for which $\arg\max_j \left[\frac{1}{L}\sum_{l=1}^{L} p_l(\omega_j|\mathbf{x})\right]$; that is, the class predicted by the majority of the classifiers. We refer to this as a consensus of certainty, and is a way of exploiting the uncorrelated variance in each of the individual classifiers.

However, in some cases it is desirable to know when there is no consensus, or more specifically when the ensemble cannot return a confident classification. Here we are not interested in knowing whether weak learners agree or disagree about the class of $\mathbf{x}$, but rather about the degree of confidence the weak learners have in assigning $\mathbf{x}$ to one of ω_j, $j \in \{1, \cdots, c\}$. The problem may be restated to ask whether $\mathbf{x}$ should belong to an "ambiguous" class or not, where ambiguousness refers to the difficulty (or lack of confidence) in classifying a sample.

1.2 Active Learning for Cost-Effective Training

Active Learning (AL) is a method of intelligently training a classifier, mitigating several drawbacks of the more standard Random Learning (RL), where samples are randomly selected for labeling [3]. RL assumes that large amounts of labeled data are already available, but for biomedical domains, manual labeling is costly and time-consuming. For example, digital images of pathology slides can be several gigabytes in size. To build a classifier to detect disease in these images, an expert pathologist needs to provide precise annotation of disease extent in the image. This results in a large training cost if RL is employed. In contrast, AL selects samples from an unlabeled pool for annotation based on the ambiguity of a sample: samples that are difficult to classify are not currently well-represented within the training set, so by targeting these samples, fewer training samples are needed to achieve high accuracy. Thus by finding only the most difficult to classify samples, we identify the most critical for labeling and inclusion in the training set.

1.3 Current Active Learning Approaches

There are several AL methods for selecting training samples [3,4,5], each relying upon a single measurement of ambiguity. The Query-By-Committee (QBC) method by Seung, et al. [5] trains a group of L weak learners, each of which votes on the class of sample $\mathbf{x}$. In the two-class case, if the sample receives approximately $\frac{L}{2}$ votes for both classes, then $\mathbf{x}$ is considered ambiguous (difficult to classify). Li, et al. [4] utilized a support-vector machine approach, whereby samples appearing close to a decision hyperplane in high-dimensional space are considered ambiguous. There is no guarantee that each of these methods will identify the same samples as "difficult to classify," since samples that are close to a decision hyperplane may still be unanimously identified as a single class by a QBC algorithm. Thus, the set of ambiguous samples may depend heavily on the AL method.

1.4 Novel Contributions of This Paper

In this paper, we present the concept of a consensus of ambiguity (CoA) whereby several measures of ambiguity are combined to identify the most difficult to classify samples from an unlabeled pool. This framework extends beyond the traditional AL methods by identifying ambiguousness explicitly rather than as a function of classification error. We define a consensus ratio that measures the degree of overlap between multiple algorithms for finding ambiguity, and we find that using multiple algorithms ensures that the overlap between methods decreases; the use of multiple algorithms ensures that only the most difficult to classify samples are detected by the algorithm.

We evaluate the efficacy of the algorithm by using the CoA-based AL method to train a probabilistic boosting tree (PBT) classifier on three separate medical image datasets. We use the performance of the PBT, measured in terms of accuracy and area under the receiver operating characteristic curve (AUC), to ensure that the training set created by CoA-based AL can yield higher performance compared to a randomly-selected training set of equal size. The three datasets considered in this work are: (1) Digitized prostate histopathology (100 images) are broken up into 12,000 image regions, each of which is classified as cancer / non-cancer using texture features. (2) 18 prostate dynamic contrast-enhanced MRI (DCE-MRI) images (256x256 pixels) are quantified using textural and functional intensity features to find cancer in a pixel-wise fashion. (3) 9,000 regions of interest (ROIs) are extracted from two large breast histopathology patient studies, with each ROI corresponding to either high or low Bloom-Richardson cancer grades. ROIs are quantified by graph-based nuclear architectural features. Each of these datasets represents different modalities, tissues, and features, but all are time-consuming and expensive to annotate; thus, we expect that AL training algorithms can reduce the expense required to obtain reliable training sets versus a random learning scheme.

2 Theory of CoA

2.1 Active Learning Strategy Overview

We denote by X a set of data containing samples $\mathbf{x} \in X$. Each sample is associated with a class label $y \in \{\omega_1, \omega_2, \cdots, \omega_c\}$. A supervised classifier is denoted

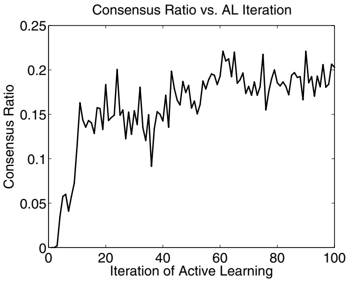


Fig. 1. Plot of the consensus ratio $\mathcal{R}$ as a function of t, for $t \in \{1, 2, \cdots, 100\}$. After $t = 50$, the consensus ratio plateaus at approximately 0.2. This indicates that there is relatively little consensus between three AL methods: Φ_1 (QBC), Φ_2 (BAY), and Φ_3 (SVD).

$\mathcal{C}(\mathbf{x}) \in \{\omega_1, \omega_2, \cdots, \omega_c\}$. The classifier returns a hypothesis for a sample and is trained on a training set S^{tr} and tested on an independent testing set. The goal of the AL algorithm is to build S^{tr} from a set of unlabeled samples in X. To do this, a training function $\Phi(\mathbf{x})$ returns a measure of ambiguity for $\mathbf{x}$.

Definition 1. *A sample $\mathbf{x} \in X$ is considered ambiguous if $a < \Phi(\mathbf{x}) < b$, where a, b are lower and upper thresholds for Φ, respectively.*

2.2 Consensus of Ambiguity: Definition and Properties

The CoA approach employs multiple algorithms, $\Phi_1, \Phi_2, \cdots, \Phi_M$, each of which returns a corresponding set of ambiguous samples $S_1^{\text{E}}, S_2^{\text{E}}, \cdots, S_M^{\text{E}}$.

Definition 2. *Given nonempty sets of ambiguous samples, S_i^E, $i \in \{1, \cdots, M\}$, the consensus ratio is defined as $\mathcal{R} = \frac{U}{V}$, where $U = |\bigcap_{i=1}^{M} S_i^E|$ and $V = |\bigcup_{i=1}^{M} S_i^E|$.*

Proposition 1. *Given nonempty sets of ambiguous samples, S_i^E, where $i \in \{1, \cdots, M\}$, $\mathcal{R} = 1$ indicates perfect consensus and $\mathcal{R} = 0$ indicates no consensus across Φ_i.*

Proof. In the case of absolutely no consensus (i.e. no samples are considered ambiguous by all M algorithms), then $\bigcap_{i=1}^{M} S_i^{\text{E}} = \emptyset$, so $\mathcal{R} = 0$. Conversely, when Φ_i, $i \in \{1, \cdots, M\}$ are in perfect agreement (every algorithm identifies exactly the same samples as ambiguous), then $S_1^{\text{E}} = \cdots = S_M^{\text{E}}$, so $\bigcap_{i=1}^{M} S_i^{\text{E}} = \bigcup_{i=1}^{M} S_i^{\text{E}}$ and $\mathcal{R} = 1$. □

Property 1. When $\mathcal{R} \approx 0$, there is low consensus and high variance among Φ_i, $i \in \{1, \cdots, M\}$, indicating that any agreement among the algorithms will be highly informative and suggesting a benefit to using a consensus approach. Figure 1 shows a graph of $\mathcal{R}$ as a function of t, which identifies the iterations of the AL algorithm. Beginning with $t = 0$, the AL algorithm grows a training set by selecting and labeling ambiguous samples and adding them to the training set. The process iterates for $t \in \{1, \cdots, 100\}$ times in this experiment. Three different AL algorithms were used: QBC, BAY, and SVD (Section 3.2). After 50 iterations, $\mathcal{R}$ levels off at approximately 0.2, indicating that there is little consensus among the methods. Thus, a consensus algorithm is likely to be informative.

Definition 3. *A sample $\mathbf{x} \in X$ will be considered strongly ambiguous if $\mathbf{x} \in \widehat{S}^E = \bigcap_{i=1}^{M} S_i^E$; that is, if the sample is designated as ambiguous by all Φ_i for $i \in \{1, \cdots, M\}$.*

Definition 3 is a version of strong ambiguity wherein all M algorithms must select the sample. It is possible that, on any particular AL iteration, no samples will satisfy this criteria. Definition 3 can easily be modified to include samples selected by a majority of algorithms, or any sample identified by more than one algorithm, and so on.

Proposition 2. *As the number of algorithms Φ_i, $i \in \{1, \cdots, M\}$, being combined increases, the consensus ratio $\mathcal{R}$ will monotonically decrease.*

Proof. An added algorithm, denoted Φ_{M+1}, identifies a set of samples denoted S_{M+1}^{E}. If S_{M+1}^{E} is a subset of the current set of ambiguous samples, $\bigcup_{i=1}^{M} S_i^{\mathrm{E}}$, then the denominator of $\mathcal{R}$ does not change since the union will not increase in size. The denominator of $\mathcal{R}$ will decrease, since any elements in $\bigcap_{i=1}^{M} S_i^{\mathrm{E}}$ that are not found in S_{M+1}^{E} will be removed in the new intersection, $\bigcap_{i=1}^{M+1} S_i^{\mathrm{E}}$. Thus $\mathcal{R}$ will decrease in value.

However, if S_{M+1}^{E} contains unique samples not in the current ambiguous sample set, the union will increase in size; that is, $|\bigcup_{i=1}^{M} S_i^{\mathrm{E}}| < |\bigcup_{i=1}^{M+1} S_i^{\mathrm{E}}|$. Thus the denominator of $\mathcal{R}$ will increase. The numerator of $\mathcal{R}$ will not change, since any samples in S_{M+1}^{E} that are not in $\bigcap_{i=1}^{M} S_i^{\mathrm{E}}$ will be removed in the new intersection, $\bigcap_{i=1}^{M+1} S_i^{\mathrm{E}}$. In this case, $\mathcal{R}$ will decrease. □

Property 2. Adding additional algorithms to the ensemble, will decrease or maintain $\mathcal{R}$. By Property 1, ensembles with a low consensus ratio $\mathcal{R}$ ensure that only samples with a very high degree of ambiguity will be identified. Thus increasing M will ensure that only extremely ambiguous samples are included in $\widehat{S}^{\mathrm{E}}$. However, if $S_{M+1}^{\mathrm{E}} \cap \widehat{S}^{\mathrm{E}} = \emptyset$, then no samples will be considered strongly ambiguous.

3 Experimental Setup

3.1 Overview of Datasets

Experiment 1 - Prostate cancer on digitized histopathology: Over a million annual prostate biopsies are performed in the US, each of which must

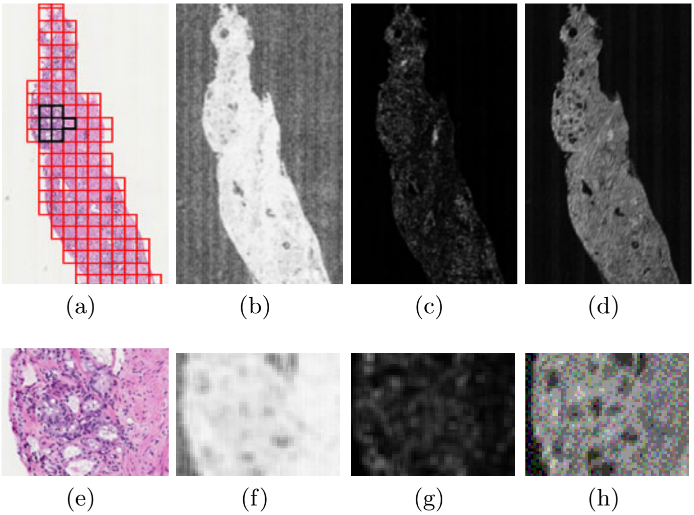


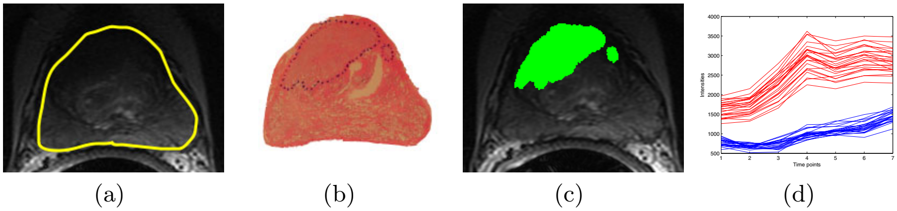
Fig. 2. Image data from Experiment 1. The original image (a) has a red 30-pixel square grid superimposed, with cancer labeled in black. Texture images are extracted corresponding to first-order greylevel statistics (b), second-order Haralick co-occurrence features (c), and Gabor steerable filter features (d). Shown in the second row (e)-(h) are magnified regions of the cancer region in each image.

be analyzed manually under a microscope [7]. A quantitative system capable of automatically detecting disease can greatly increase the speed and accuracy with which patients are diagnosed for cancer. Digitized glass slides can be over 2 GB in size (several million pixels), with benign and cancer regions appearing close to one another, and so annotation of these samples is difficult. The objective of this experiment is to apply CoA-based AL to build a classifier able to distinguish between cancerous and non-cancerous patches of biopsy tissue.

Biopsy samples are stained with Hematoxylin and Eosin (H & E) to visualize cell cytoplasm and nuclei and digitized using a whole-slide digital scanner. For each image, a 30x30 pixel grid is superimposed on the tissue, generating regions of interest (ROIs) of prostate tissue. In previous work [8], we have identified 14 texture features that can easily distinguish between cancer and non-cancer regions of tissue on a pixel-wise basis. These features include: (1) First-order gray-level statistics quantify simple statistics calculated from pixel values in the images [8]. (2) Second-order Haralick features [9] are based on the co-occurrence of pixel values, and are calculated over each ROI. (3) Gabor filter features, also known as steerable filters, operate at a specific orientation and spatial frequency to yield a filter response from the image. Each of the 14 discriminating features is extracted from the image, and the modal value for each 30-by-30 ROI is used as its feature value. 100 images are used to generate 12,000 ROIs which are classified as cancer or non-cancer tissue.

Experiment 2 - Prostate cancer on DCE-MRI: In addition to biopsy, *in vivo* imaging, particularly magnetic resonance imaging (MRI), can be mined for quantitative diagnostic information [10,11]. Dynamic Contrast Enhanced (DCE) MRI is a technique whereby a contrast agent is injected into a patient with MR images taken at specific time points. The contrast agent is taken up and removed from different tissues at different rates, indicating the presence of disease at a pixel-wise level. A classification system for this modality could be used for automated *in vivo* screening for cancer and treatment, but labeled samples are difficult to obtain since cancer cannot be annotated directly on the MRI. Histopathology is used to find cancer ground truth, which is mapped onto the MR images.

We apply CoA-based AL to a dataset of 6 patients with confirmed prostate cancer on needle biopsies. Prior to radical prostatectomy, MR imaging was performed using an endorectal coil in the axial plane and included T2-w and DCE protocols. Prostatectomy specimens were later sectioned and stained with H & E. An expert pathologist annotated the spatial extent of prostate cancer on the whole-mount prostatectomy sections, and identified 18 corresponding histopathology and MRI sections. A multimodal registration scheme, COLLection of Image-derived Non-linear Attributes for Registration Using Splines (COLLINARUS) [12], was used to register histology sections onto the corresponding MRI data, thus mapping the cancer ground truth onto the MR images. Structural information from T2-w MRI and functional intensity information from DCE MRI are combined to distinguish between cancer and non-cancer pixels.



(a) (b) (c) (d)

Fig. 3. Examples of data from Experiment 2. Shown are (a) T2-w MRI image with the prostate boundary in yellow, (b) the corresponding histopathology slice with cancer mapped in blue, and (c) the cancer extent mapped onto the T2-w MRI after registration via COLLINARUS [12]. Also shown are (d) intensity vs. time curves for dynamic contrast; blue curves represent pixel locations in benign tissues, while red curves are inside cancer ground truth ((c)).

Experiment 3 - Breast cancer on digitized histopathology: Breast cancer is the second-leading cause of cancer death in women in the United States [7]. Mammogram screening followed by a biopsy is the current standard for definitive diagnosis. Similar to the motivation in Experiment 1, an automated image analysis system can assist pathologists in detecting and diagnosing breast cancer.

Images of H & E stained breast biopsies are classified between low and high Bloom-Richardson grades of breast cancer. Two patient studies were used to

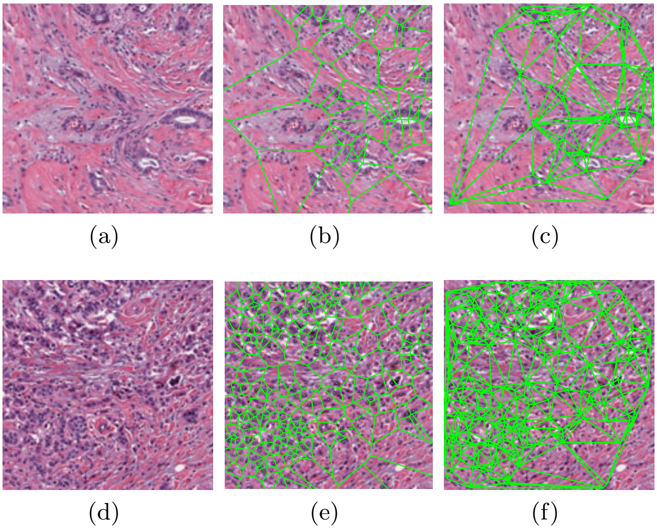


Fig. 4. Examples of image data from Experiment 3, where we distinguish low-grade breast cancer tissue ((a)-(c)) from high-grade tissue ((d)-(f)). Nuclei are detected from breast biopsy tissue (a), (d) and used to generate graphs such as the Voronoi tesselation (b), (e) and Delaunay triangulation (c), (f). Features from these graphs are used to quantify each image patch.

generate 9,000 ROIs of homogeneous tissue measuring 500x500 pixels each. We calculate features based on the architecture of the cell nuclei, in accordance with the major indicators of breast cancer grade. Color deconvolution is used to transform the RGB color space of the image into an alternate three-color space to separate out the hematoxylin, eosin, and white background of the image [13]. Using the deconvoluted image, the centroids of cell nuclei are detected, which are used to construct a series of graphs based on the Voronoi tesselation, Delaunay triangulation, and a minimum spanning tree. From each of these, a set of quantitative features is extracted to characterize the cell architecture [13]. Each ROI is classified as high or low Bloom-Richardson grades of cancer, where ground truth is determined by a pathologist.

3.2 Comparison of AL Methods

Query-By-Committee (QBC): QBC [5] involves a group of L weak classifiers that produce votes for the class of an unlabeled sample $\mathbf{x}$. Samples with approximately $\frac{L}{2}$ votes are considered difficult to classify. The output of $\Phi_1(\mathbf{x})$ is the number of votes for the target class, and a, b represent the minimum and maximum votes, respectively. A total of $L = 10$ Random Forests were generated using C4.5 decision trees [2,1] with threshold values of $a = 4$ and $b = 6$.

Bayes Likelihood (BAY): Bayes Theorem [14] models the likelihood of observing a class based on the feature values of sample $\mathbf{x}$. A probability density function is created for each of K features, where $p_k(\omega_j|\mathbf{x})$ denotes the likelihood that $\mathbf{x}$ belongs to class ω_j given feature k. Samples for which $p_k(\omega_j|\mathbf{x}) \approx 0.5$ are considered ambiguous. The output of $\Phi_2(\mathbf{x})$ is $\frac{1}{K}\sum_{k=1}^{K} p_k(\omega_1|\mathbf{x})$ where ω_1 is the target (cancer) class. Threshold parameters were set to $a = 0.4$ and $b = 0.6$.

Support Vector Distance (SVD): Support Vector Machines (SVMs) [15] create a high-dimensional projection of feature data, in which a decision hyperplane is created via training. Samples are classified by finding the position relative to the hyperplane. The output of $\Phi_3(\mathbf{x})$ is the signed distance between $\mathbf{x}$ and the hyperplane, where the sign indicates class membership. Parameters a and b define the distances within which a sample is considered ambiguous. We set a and b to $\pm 10\%$ of the maximum distance from the support vector.

3.3 Probabilistic Boosting Tree Classification Algorithm

CoA-based AL was used to train a probabilistic boosting tree (PBT) [16]. The PBT combines AdaBoost [17] and decision trees [1], iteratively generating a tree where each node is boosted with L weak classifiers and whose output is a likelihood for the class of sample $\mathbf{x}$. The PBT algorithm was chosen as a classifier that is different from the methods used in each of the AL algorithms described above. At each iteration of the active learning algorithm, $t \in \{1, 2, \cdots, 100\}$, ambiguous samples found by the CoA ensemble are sampled to obtain equal numbers of samples from both classes [6], which are used to train the PBT. For our experiments, each iteration added two samples (one from each class) to the growing training set. Evaluation on an independent testing set is done via area under the receiver operating characteristic curve (AUC) and accuracy.

4 Results and Discussion

Shown in Figure 5 are examples of two datasets, prostate histopathology (top row) and DCE-MRI (bottom row), used in this study. In the left column (Figures 5 (a), (d)) are the original images with the cancerous region delineated in a black contour, while the results of classification with RL training are shown in the middle column (Figures 5 (b), (e)) and training with CoA-based AL are shown in the right column (Figures 5 (c), (f)). The images were obtained when the AL algorithm had run for $t = 50$ iterations.

For histopathology, brighter regions indicate higher likelihood of cancer. The RL-trained classifier identifies the majority of patches as cancer yielding a high false-positive count, while the CoA-trained classifier is able to discriminate between obviously benign regions and cancerous areas. Note that we are not commenting here on the accuracy of the final classifier, but on the performance of

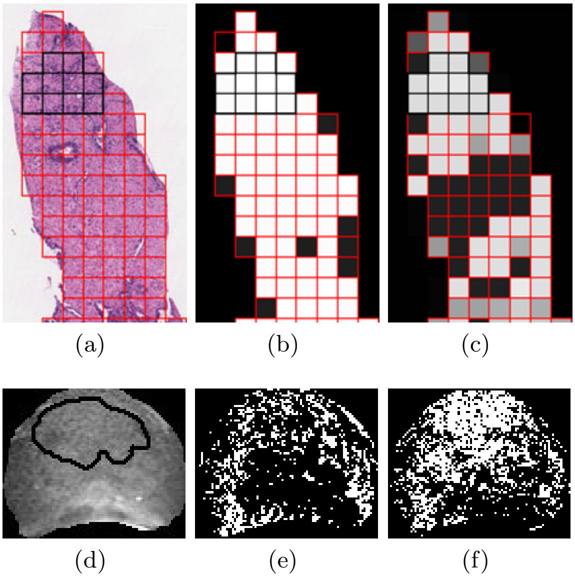


Fig. 5. Examples of images taken from the prostate histopathology (a) and DCE-MRI (d) datasets, with cancer regions indicated by black contours. Also shown are the corresponding classification results of the PBT, when using training sets built via RL ((b), (e)) and CoA-based AL ((c), (f)). Images were obtained at AL iteration $t = 50$.

one training method with respect to another. For the DCE images, images were thresholded at a likelihood of 75%. Here, the RL-trained classifier yields false-negatives with a small set of pixels classified as cancer, while the CoA-trained classifier correctly classifies many pixels near the ground truth. Again, this indicates that – given the limitations on labeling biomedical images – CoA yields better results than random training on a limited number of training samples.

The accuracy and AUC of the PBT are plotted against the AL iteration $t \in \{1, \cdots, 100\}$ in Figure 6. Shown are the results for the classifier trained using the CoA algorithm (red solid) as well as random learning (blue dotted) and each of the three AL strategies: QBC (green dot), BAY (cyan solid), and SVM (magenta dash). Each location on the independent axis indicates a training set size (increasing from left to right); we can see that for the majority of training set sizes, all of the AL-trained classifiers yield better accuracy and AUC than random learning. Additionally, AL requires fewer samples to reach that desired performance compared with RL. We note that the individual AL algorithms do not necessarily perform better than the CoA approach in terms of classifier performance, but this is not an unexpected result. The goal of using the CoA algorithm is to prune down the number of samples deemed "eligible" at each stage; we see that by constraining our search in this way, we have a smaller pool from which to choose labeled samples, while keeping performance the same as an individual algorithm (which has a much wider set of "eligible" samples).

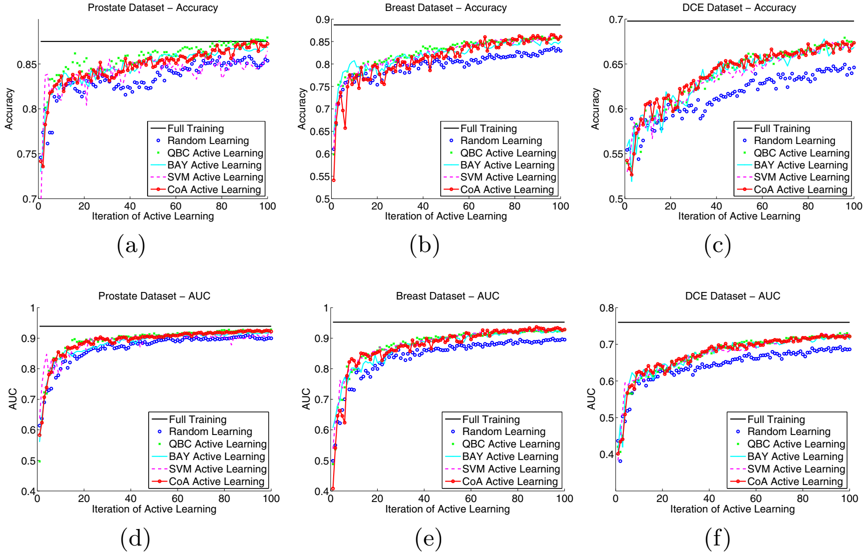


Fig. 6. Plots of the accuracy and AUC obtained by the PBT using the training derived from CoA Active Learning method (red solid line), which combines three AL schemes (QBC, BAY, and SVD), and Random Learning (blue dotted line). Shown are results for the dataset of 12,000 prostate histopathology ROIs ((a), (d)), 28,000 prostate DCE-MRI pixel samples ((c), (f)), and 9,000 breast histopathology ROIs ((b), (e)).

5 Concluding Remarks

In this paper, we presented a CoA framework for identifying ambiguousness in an unlabeled pool of data. The CoA approach exploits variance between different ambiguity measurements. A consensus ratio determines the amount of variance between multiple ambiguity methods, and by combining these algorithms, this ratio decreases. This ensures that only the most ambiguous samples are selected from the unlabeled data. Finally, we applied CoA to the problem of Active Learning (AL), where ambiguous samples are selected for training a classifier. For medical image datasets (which are time-consuming and expensive to annotate), the CoA-trained classifier yields higher accuracy and AUC than RL for similar training set sizes.

We observe similar classification performance using CoA versus individual AL training schemes. However, the low consensus ratio indicates that each training algorithm is selecting mostly unique samples. Since our goal is to improve training efficiency, we wish to explore evaluation measures besides classifier performance. For example, it is possible that samples selected by one AL scheme are more difficult to annotate than those selected by another, or have significantly different feature distributions. If so, we may be able to derive an evaluation metric that is divorced from classifier performance that is able to identify the most efficient training algorithm.

References

1. Quinlan, J.R.: Decision trees and decision-making. IEEE Trans. Syst. Man Cybern. 20(2), 339–346 (1990)
2. Brieman, L.: Random Forests. Machine Learning 45(1), 5–32 (2001)
3. Cohn, D., Ghahramani, Z., Jordan, M.I.: Active Learning with Statistical Models. J. of Art. Intel. Res. (4), 129–145 (1996)
4. Li, M., Sethi, I.K.: Confidence-based active learning. IEEE Trans. Patt. Anal. Mach. Intel. 28(8), 1251–1261 (2006)
5. Seung, H.S., Opper, M., Sompolinsky, H.: Query by committee. In: 5th Annual ACM Workshop on Computational Learning Theory, pp. 287–294. ACM, New York (1992)
6. Doyle, S., Madabhushi, A., Feldman, M., Tomaszewski, J., Monaco, J.: A Class Balanced Active Learning Scheme that Accounts for Minority Class Problems: Applications to Histopathology. In: OPTIMHisE Workshiop (in conjunction with MICCAI), pp. 19–30 (2009)
7. American Cancer Society. Cancer Facts & Figures 2010. American Cancer Society, Atlanta (2010)
8. Doyle, S., Feldman, M., Tomaszewski, J., Madabhushi, A.: A Boosted Bayesian Multi-Resolution Classifier for Prostate Cancer Detection from Digitized Needle Biopsies. IEEE Transactions on Biomedical Engineering (accepted)
9. Haralick, R.M., Shanmugan, K., Dinstein, I.: Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics SMC 3, 610–621 (1973)
10. Madabhushi, A.: Digital Pathology Image Analysis: Opportunities and Challenges. Imaging in Medicine 1(1), 7–10 (2009)
11. Viswanath, S., Bloch, B.N., Rosen, M., Chappelow, J., Rofsky, N., Lenkinski, R., Genega, E., Kalyanpur, A., Madabhushi, A.: Integrating Structural and Functional Imaging for Computer Assisted Detection of Prostate Cancer on Multi-Protocol in vivo 3 Tesla MRI. In: SPIE Medical Imaging, vol. 7260 (2009)
12. Chappelow, J., Madabhushi, A., Bloch, B.: COLLINARUS: Collection of image-derived non-linear attributes for registration using splines. In: Proc. SPIE: Image Processing, vol. 7259, San Diego, CA, USA (2009)
13. Basavanhally, A.N., Ganesan, S., Agner, S., Monaco, J., Feldman, M., Tomaszewski, J., Bhanot, G., Madabhushi, A.: Computerized Image-Based Detection and Grading of Lymphocytic Infiltration in HER2+ Breast Cancer Histopathology. IEEE Transactions on Biomedical Engineering 57(3), 642–653 (2010)
14. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley Interscience, New York (2001)
15. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning 20, 273–297 (1995)
16. Tu, Z.: Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In: 10th IEEE International Conference on Computer Vision, pp. 1589–1596 (2005)
17. Freund, Y., Schapire, R.E.: Experiments with a New Boosting Algorithm. In: 13th International Conference on Machine Learning, pp. 148–156 (1996)

Semi-supervised Learning of Sparse Linear Models in Mass Spectral Imaging

Fabian Ojeda[1,⋆], Marco Signoretto[1], Raf Van de Plas[1,3], Etienne Waelkens[2,3], Bart De Moor[1,3], and Johan A.K. Suykens[1]

[1] ESAT-SCD-SISTA, Department of Electrical Engineering, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium
[2] Laboratory for Phosphoproteomics, Katholieke Universiteit Leuven, O & N, Herestraat 49, B-3000 Leuven, Belgium
[3] ProMeta, Interfaculty Centre for Proteomics and Metabolomics, Katholieke Universiteit Leuven, O & N 2, Herestraat 49, B-3000 Leuven, Belgium
{fabian.ojeda,marco.signoretto,raf.vandeplas,bart.demoor, johan.suykens}@esat.kuleuven.be,
etienne.waelkens@med.kuleuven.be
http://www.esat.kuleuven.be/sista

Abstract. We present an approach to learn predictive models and perform variable selection by incorporating structural information from Mass Spectral Imaging (MSI) data. We explore the use of a smooth quadratic penalty to model the natural ordering of the physical variables, that is the mass-to-charge (m/z) ratios. Thereby, estimated model parameters for nearby variables are enforced to smoothly vary. Similarly, to overcome the lack of labeled data we model the spatial proximity among spectra by means of a connectivity graph over the set of predicted labels. We explore the usefulness of this approach in a mouse brain MSI data set.

Keywords: MSI, sparsity, ordered variables, spatial information, smoothing penalty, graph Laplacian, convex optimization, regularization.

1 Introduction

Mass spectral imaging (MSI) is a developing technology that allows the detection of biomolecules such as proteins, peptides, and metabolites from organic tissue while retaining the spatial information intact [1]. Thus, MSI enables the study of the spatial tissue distribution for any detectable molecule that falls within a specified molecular mass range [2]. A typical MSI experiment consists of a grid of measurement locations or pixels covering the tissue section, with an individual mass spectrum attached to each pixel. The resulting data structure can be considered as three-dimensional array or tensor with two spatial dimensions (h and w) and one mass-over-charge (m/z) dimension as shown in Fig. 1.

These characteristics pose challenges in the statistical analysis of MSI data. The high molecular specificity of MSI on one hand, delivers huge dimensional

⋆ Corresponding author.

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 325–334, 2010.

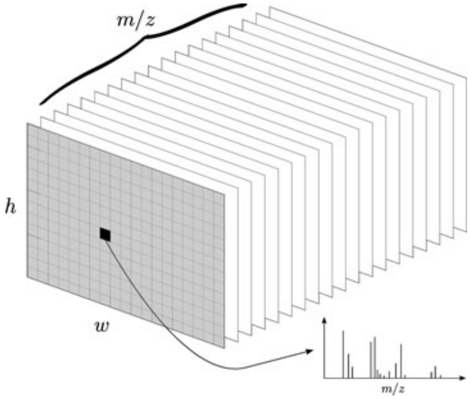


Fig. 1. A schematic representation of the MSI data structure. Individual mass spectra are collected from the tissue area of interest retaining their spatial relationships (h,w). The data is collected into a three-mode array where each *slide* corresponds to a particular (m/z) value and every point in the grid is attached to a spectrum.

data sets with thousands of measured variables that usually exceed the number of spectra (observations), often limited to a few hundreds. On the other hand, the spatial coordinates (h, w) associated to each spectrum define areas of interest and thus should be not neglected. In practice several methods have been already applied to MSI data set including but not limited to principal component analysis [3], clustering and multivariate analysis [4], and supervised classification [5],[6]. Besides the low number of observations, only a small fraction of them is labeled. This hinders many statistical methods and further limits the validation of the obtained results. Manual labeling requires dedicated expertise which can be time consuming, costly and in some cases inaccurate.

In the present article we aim to address most of the aforementioned issues. In a first step, we start from regularized models that impose sparsity in the solution of coefficients. In the problem of interest variables admit a natural ordering due to their physical meaning. Therefore we enforce that the estimated coefficients of nearby variables should smoothly vary in terms of m/z. Unlike the so called *fused lasso* [7] where the absolute value of the differences is used, we employ a smooth quadratic penalty. Furthermore, to overcome the lack of labeled observations we exploit the prior assumption that nearby spectra are likely to have the same label. This is a meaningful assumption for many type of data: for instance a tumor is more likely to affect nearby cells than erratically affect disconnected regions of tissue. Our approach encodes the spatial proximity among spectra by means of a graph and hence can be seen as a semi-supervised method. The resulting proposed model is shown to be equivalent to a *lasso* formulation and therefore can be efficiently solved via the LARS (Least Angle Regression) [8] algorithm. Each component in our optimization problem clearly embodies the structural information of MSI data, whereas regularization parameters trade off the complexity of the model in terms of sparsity, smoothness and unlabeled samples.

This paper is organized as follows. Section 2 introduces the notions about regularized linear models and notation with respect to MSI data. The general concept of encoding structural information via the graph Laplacian is presented in Section 3, while Section 3.1 deals in detail with the modeling of the ordering of the m/z variables and the resulting optimization problem. Section 3.2, elaborates on the encoding of spatial information using unlabeled samples and states the final proposed approach. Preliminary results on a mouse brain MSI data set are given in Section 4. Comparisons to related algorithms are reported along with visualization and interpretation of the obtained results.

2 Notation and Preliminaries

The MSI data set can be represented by a collection of n observations (spectra) measured over p variables (mass-to-charge ratios). The set of labeled spectra is $\mathcal{D}_\ell = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, with $\boldsymbol{x}_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$, where y_i is the associated label to spectra $\boldsymbol{x}_i \in \mathbb{R}^p$. Denote by x_i^j the j-th component of $\boldsymbol{x}_i \in \mathbb{R}^p$, therefore $\boldsymbol{x}^j = (x_1^j, x_2^j \ldots, x_i^j, \ldots, x_n^j)^\top$ indicates the vector of measurements of a single variable. We deal with the problem of predicting the response y, from a corresponding observation $\boldsymbol{x}$. In this setting we consider the standard linear regression model

$$y_i = \sum_{j=1}^{p} \beta_j x_i^j + \varepsilon_i \quad , \tag{1}$$

with errors ε_i. The variables are assumed to be standardized and the output to be centered. The vector of coefficients $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \ldots, \hat{\beta}_p)^\top \in \mathbb{R}^p$ is usually obtained by penalized empirical risk minimization:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 + \lambda P(\boldsymbol{\beta}) \quad . \tag{2}$$

Common examples of penalized models are ridge regression with $P(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|^2$, or the *lasso* with $P(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$. The lasso penalty encourages sparse solutions while ridge regression keeps all the coefficients in the model. In general, a priori assumptions encoded via $P(\cdot)$ are needed to make the problem well-posed. In the following sections we aim to model the specific features of MSI data by translating them into useful structural information in the general optimization problem described in (2).

3 Structure Encoding via the Graph Laplacian

In order to incorporate structural information in our model fitting approach, we consider an undirected connectivity graph $\mathcal{G} = (V, E)$, where V is the set of nodes and E the set of edges. An edge between given nodes u and v ($u \sim v$) exists

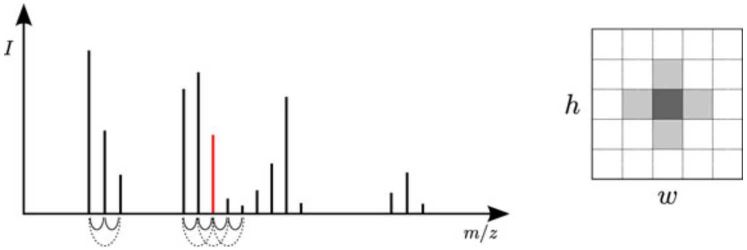


Fig. 2. *Left*: First order (solid) and second order (dashed) connectivity structure over the set of the variables. We consider first order connectivity to impose local smoothness on the coefficients. *Right*: *Cross-like* spatial neighborhood imposed over the set of spectra.

if the entities represented by u and v are linked. Denoting d_u as the degree of a node u, the normalized Laplacian matrix L associated to the graph $\mathcal{G}$ is given by [9]

$$L(u,v) = \begin{cases} 1, & \text{if} \quad u = v \quad \text{and} \quad d_u \neq 0 \ , \\ -1/\sqrt{d_u d_v}, & \text{if} \quad u \sim v \ , \\ 0, & \text{otherwise} \ . \end{cases} \tag{3}$$

The Laplacian is a symmetric semi-positive definite matrix which can be interpreted as an operator on functions of the type $\boldsymbol{f} : V \to \mathbb{R}$ namely vectors indexed by elements of V. It can be shown that [9]

$$\boldsymbol{f}^\top L \boldsymbol{f} = \sum_{u \sim v} \left(\frac{f_u}{\sqrt{d_u}} - \frac{f_v}{\sqrt{d_v}} \right)^2 \ , \tag{4}$$

and hence the quadratic term on the left-hand side of (4) can be used to define a penalty enforcing smooth variation over neighboring nodes. We use this fact to incorporate structural information of MSI into the learning framework.

3.1 Encoding Ordered Variables

In order to account for the natural ordering of the m/z measurements, we impose a graph $\mathcal{G}^p$ over the set of variables. The set of nodes are associated to the p input variables $\boldsymbol{x}^j, j = 1, ..., p$, thus modeling neighboring variables via the Laplacian matrix of the graph. The structure imposed can is visualized in the left panel in Fig.2, where every m/z variable $\boldsymbol{x}^j$ is connected to the preceding $\boldsymbol{x}^{j-1}$ and the subsequent $\boldsymbol{x}^{j+1}$. One might also consider second order relationships and so forth. By defining $L_\beta \in \mathbb{R}^{p \times p}$ as the Laplacian over the set of variables (cf. (3)) and considering the squared norm in (4) for $\boldsymbol{\beta}$, our regularized optimization problem takes then the form:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \boldsymbol{\beta}^\top L_\beta \boldsymbol{\beta} \ , \tag{5}$$

with $\lambda_1, \lambda_2 > 0$. While the second term enforces sparsity on the $\boldsymbol{\beta}$, the last term smooths the solution of $\boldsymbol{\beta}$ on the network. This is similar to the formulations in [10] and [11]. In the case that no structure is assumed in the network, that is taking $L_\beta = I$, the optimization problem resorts to the elastic net approach [12].

3.2 Encoding Prior Spatial Information

Along the same lines of reasoning, we aim at imposing also a smooth structure on the predicted labels $\hat{y}_k,\ k = 1, ..., n_s$. The spatial distribution of the spectra in the square grid (see Fig.1) suggests that nearby spectra should correspond either to the same tissue area or, might represent connectivity tissues. In essence our goal is to extend the framework exposed in the previous section by incorporating additional information about the spatial structure of the MSI data. In order to get an empirical estimate of spectra distribution we make use of unlabeled examples [13]. Denoting by $\hat{\boldsymbol{y}} = (\hat{y}_1, \ldots, \hat{y}_{n_s})^\top$ the vector of predicted responses and, assigning each $\hat{y}_k$ to a node in a graph $\mathcal{G}^s$, we construct the corresponding Laplacian matrix $L_s \in \mathbb{R}^{n_s \times n_s}$ using (3). The entries of $L_s(h, w)$ are defined according to the *cross-like* neighborhood pattern shown on the right hand side of Fig.2. Likewise, we consider a similar quadratic form for the predicted responses as in (4), that is $\hat{\boldsymbol{y}}^\top L_s \hat{\boldsymbol{y}}$, which is bounded by an user specific parameter $\xi > 0$. Including this constraint into our optimization problem we have

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \boldsymbol{\beta}^\top L_\beta \boldsymbol{\beta} \tag{6}$$

$$\text{s.t.} \quad \hat{\boldsymbol{y}}^\top L_s \hat{\boldsymbol{y}} \leq \xi \tag{7}$$

$$\hat{y}_k = \sum_{j=1}^{p} \beta_j x_k^j,\ k = 1, \ldots, n_s\ . \tag{8}$$

Expressing the vector of equality constrains in matrix form $\hat{\boldsymbol{y}}_s = X_s\boldsymbol{\beta}$, and replacing this term in the inequality constraint we get $\boldsymbol{\beta}^\top (X_s)^\top L_s (X_s)\boldsymbol{\beta} = \boldsymbol{\beta}^\top G_s \boldsymbol{\beta}$. By introducing a Lagrange multiplier $\lambda_3 > 0$ for the latter constraint, we write (6) as the following unconstrained optimization problem

$$\hat{\boldsymbol{\beta}} = \arg\min_{\beta} \|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \boldsymbol{\beta}^\top L_\beta \boldsymbol{\beta} + \lambda_3 \boldsymbol{\beta}^\top G_s \boldsymbol{\beta}\ . \tag{9}$$

By grouping the quadratic terms of $\boldsymbol{\beta}$ and defining $H_{\lambda_3} = L_\beta + \frac{\lambda_3}{\lambda_2} G_s$, we can further cast the optimization problem into a *lasso* type formulation

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \arg\min_{\beta} \|\boldsymbol{y} - X\boldsymbol{\beta}\|_2^2 + | + \|\mathbf{0}_{p\times 1} - \sqrt{\lambda_2} H_{\lambda_3}^{1/2} \boldsymbol{\beta}\|_2^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j|\ , \\ &= \left\| \begin{bmatrix} \boldsymbol{y} \\ \mathbf{0}_{p\times 1} \end{bmatrix} - \begin{bmatrix} X \\ \sqrt{\lambda_2} H_{\lambda_3}^{1/2} \end{bmatrix} \boldsymbol{\beta} \right\|_2^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j|\ . \end{aligned} \tag{10}$$

This modified problem has dimensions $(n + p) \times p$, and can be solved using the LARS (Least Angle Regression) algorithm [8] up to fixing λ_2 and solving the regularization path for the constrained version using a bound $|\boldsymbol{\beta}|_1 \leq \tau$ instead of λ_1. This equation is our final proposed method to incorporate spatial information and to impose neighboring structure in the ordered variables.

By setting $\lambda_3 = 0$, we can relate our method to similar existing approaches. For instance, the Elastic net [12] is obtained by setting $L_\beta = I$. The network constrained-regularization (NET)[10] and the multiple NET [11] are not restricted to ordered variables and instead they impose prior groupings through graphs. The fused lasso algorithm in [7] penalized the absolute difference between adjacent coefficients whereas the the *group lasso* [14] assumes in advance groups of variables.

4 Experimental Results

In this section we explore the usefulness and applicability of the proposed method to include the structural information of MSI data. While numerical validation of the obtained results is assessed via 10-fold cross-validation, visual interpretation and comparison appear more intuitive by translating the results into exploratory ion images. This visual aid compensates the limited availability of ground truth information.

4.1 Data Set

The data set, acquired at University Hospital Leuven, comes from a sagittal section of mouse brain [15]. The spatial grid covering the tissue has 51 × 34 measurement locations (i.e. 1734 pixels). Each measurement spans a mass range from 2800 to 25000 Dalton in 6490 mass-over-charge (m/z) bins. Therefore, the data structure contains 1734 mass spectra measuring 6490 m/z variables per spectrum. Partial labeling information of 279 spectra is provided by a pathologist corresponding to four anatomical regions within the tissue. The labeled regions are the cerebellar cortex (cc), Ammon's horn in the hippocampus (ca), the caudaputamen (cp), and the lateral ventricle (vl) area. Figure 3(a) depicts the four partially labeled regions overlaid on a gray level microscopic image of the mouse brain section. The set of spectra is normalized with respect to the total ion current and is baseline corrected.

4.2 Numerical Results

In order to set suitable values for the three regularization parameters, we first define a grid of values over the parameters λ_2 and λ_3. Secondly, for every pair of values we approximate the regularization path for the parameter τ (associated to λ_1) and pick the best combination via 10-fold cross-validation. In Table 1, we report the results of the proposed approach among pairwise classes. Chosen values for the regularization parameters are reported along with the average 10-fold

Table 1. Multi-class one-vs-one results of the proposed approach. Regularization parameters associated to the quadratic penalties (λ_2, λ_3) are chosen from the grid $[10^{-3},\ 10^{-2},\ 10^{-1},\ 1,\ 10,\ 100]^2$. The regularization path for parameter τ (bound on the L_1 norm) associated to λ_1 is optimized via 10-fold crossvalidation on the labeled data.

Classes	τ^*	λ_2^*	λ_3^*	Non-zero β	10-fold mse	10-fold accuracy
cc vs *ca*	0.165	100	0.001	64	1.0529 (0.5482)	1 (0)
cc vs *cp*	0.213	100	0.001	106	1.3783 (0.3739)	0.9288 (0.1076)
cc vs *vl*	0.249	1	0.001	56	1.7588 (0.9709)	0.8938 (0.1719)
ca vs *cp*	0.162	1	0.001	54	2.3778 (0.9005)	0.9758 (0.0319)
ca vs *vl*	0.1640	100	0.01	102	3.2342 (1.2151)	0.9446 (0.0447)
cp vs *vl*	0.0460	10	0.1	14	5.9332 (1.6775)	0.9288 (0.0580)

Table 2. Multi-class one-vs-one results of the LASSO (lasso) and Elastic net (enet) algorithms

	lasso			enet			
Classes	Non-zero β	τ^*	10-fold accuracy	Non-zero β	τ^*	λ_2^*	10-fold accuracy
cc vs *ca*	14	0.29	1.0000 (0.0)	17	0.211	0.001	1.0000 (0.0)
cc vs *cp*	10	0.16	0.9738 (0.0532)	15	0.136	0.01	0.9905 (0.0202)
cc vs *vl*	34	0.374	0.9250 (0.1208)	17	0.145	0.01	0.9333 (0.1097)
ca vs *cp*	31	0.212	0.9740 (0.0436)	18	0.1	0.001	0.9687 (0.0477)
ca vs *vl*	11	0.085	0.9143 (0.0732)	19	0.076	0.1	0.9330 (0.0549)
cp vs *vl*	5	0.031	0.9142 (0.0739)	6	0.033	0.001	0.9123 (0.0766)

Table 3. Combined multi-class results on the MSI mouse brain rat data set

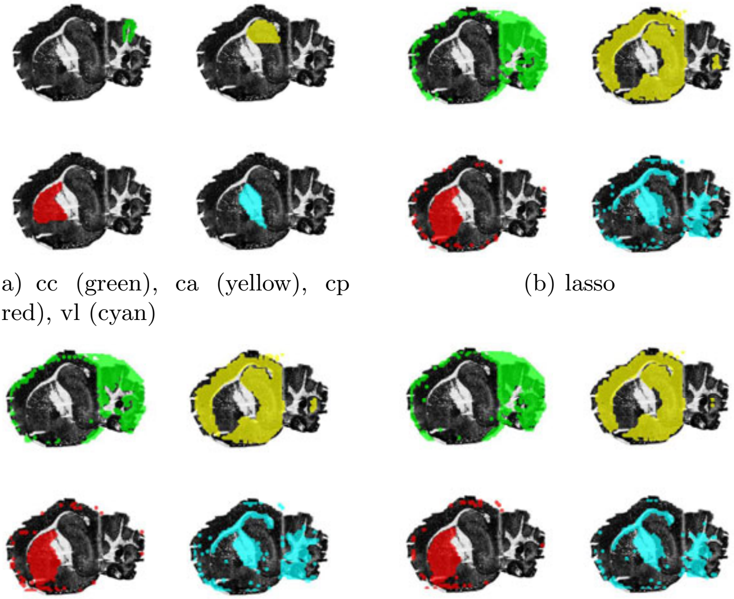
Method	Avg. Non-zero β_j	Avg. $\lvert\beta\rvert_1$	10-fold accuracy
lasso	17	1.0303	0.9453 (0.0690)
enet	16	0.9608	0.9563 (0.0515)
proposed	66	1.6647	0.9502 (0.0608)

cross-validation accuracy and the number of non-zero β coefficients. Similarly, the performance of the lasso and elastic net algorithms are reported in Table 2.

Additionally, the performance for the combined *one-versus-one* predictions is presented in Table 3. All the three methods perform slightly similar with appreciable differences in the average number of coefficients. The proposed method tends to select more coefficients due to the effect of the two square penalties.

4.3 Visualization

By translating the predicted labels back to their position in the spatial domain, one can directly assess the performance of the algorithm via visual inspection.



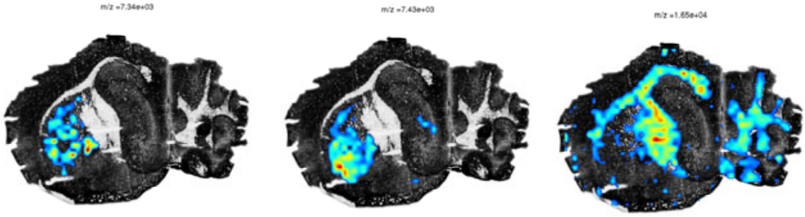
(a) cc (green), ca (yellow), cp (red), vl (cyan)

(b) lasso

(c) enet

(d) proposed

Fig. 3. Labeled areas and corresponding predicted labels by the algorithms



(a) $m/z = 7.34 \times 10^3$ (b) $m/z = 7.43 \times 10^3$ (c) $m/z = 1.65 \times 10^4$

Fig. 4. Ion image visualization for the top three selected m/z variables discriminating the (cc) and (vl) tissue regions. The first two are common to the three compared methods, whereas the third variable at $m/z = 1.65e + 4$ Da that delineates the (vl) area only appears in the proposed model.

Figure 3 displays the combined *one-vs-one* predicted labels corresponding to the compared methods. All the three models effectively separate the lateral ventricle (vl) and cauda-putamen (cp) from the surrounding tissue. The classification for the ventricle area additionally draws in the elongated corpus callosum and cerebellar nucleus regions as well, as these regions share a panel of common molecules within the measured mass range. The cerebellar cortex (cc) label exceeds its intended boundaries due to the small number of labeled spectra (21 observations). The remaining hippocampus label (ca) extends to capture the complete hippocampus and most of the remaining unlabeled areas of the tissue.

Furthermore, to visualize important selected variables, we look at the top three variables associated to the largest β coefficients. In particular we take those differentiating the lateral ventricle (vl) from the cauda-putamen (cc). In Fig.4, ion images highlight the presence of three of the top selected m/z in these two anatomical regions.

5 Conclusion

In this article we have presented a methodology to learn semi-supervised sparse linear models in MSI data. Starting from regularized learning models and structural information inherent to MSI data, we make use of the graph Laplacian to embed first, the natural ordering of the m/z variables and, secondly the spatial location of the spectra. Thereby, smooth quadratic penalties are imposed over neighboring *nodes* representing in the first case *variables* and in the second one *observations*. These penalties modify the standard learning algorithm resulting in an equivalent *lasso* formulation that can be solved efficiently. Moreover the lack of labeled data, typical of MSI experiments, is circumvented through modeling the predicted responses via the graph Laplacian. The applicability of the proposed approach is explored in a mouse brain MSI data set to distinguish amongst four anatomical regions, and it is compared to other learning models that do not, or partially, incorporate the structural information of MSI data. The presented case study shows that sparse linear models can already provide significant informative insight to assess tissue type, structure, and content. Additionally, our approach also holds value for more fundamental exploratory studies of tissue as it can highlight similarity in content between different tissue areas. Further work in this direction seems promising and should find applicability as more MSI data sets become available.

Acknowledgments. *Research Council KUL*: ProMeta, GOA Ambiorics, GOA MaNet, CoE EF/05/007 SymBioSys, START 1, Optimization in Engineering (OPTEC), IOF-SCORES4CHEM, several PhD/postdoc & fellow grants. *Flemish Government*: FWO: PhD/postdoc grants, G.0318.05 (subfunctionalization), G.0553.06 (VitamineD), G.0302.07 (SVM/Kernel), G0226.06 (cooperative systems and optimization), G0321.06 (Tensors), G.0302.07 (SVM/Kernel), G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine) research communities (ICCoS, ANMMM, MLDM); G.0377.09 (Mechatronics MPC), G.0733.09 (3UTR), G.082409 (EGFR). IWT: PhD Grants, Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&O-Dsquare, Silicos, SBO-BioFrame, SBO-MoKa, TBM-IOTA3. FOD:Cancer plans. *Belgian Federal Science Policy Office*: IUAP P6/25 (BioMaGNet, Bioinformatics and Modeling: from Genomes to Networks, 2007-2011), IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011). *EU-RTD: ERNSI*: European Research Network on System Identification; FP7-HEALTH CHeartED. FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940).

References

1. Stoeckli, M., Chaurand, P., Hallahan, D.E., Caprioli, R.M.: Imaging mass spectrometry: A new technology for the analysis of protein expression in mammalian tissues. Nature Medicine 7(4), 493–496 (2001)
2. McDonnell, L.A., Heeren, R.M.A.: Imaging mass spectrometry. Mass Spectrometry Reviews 26(4), 606–643 (2007)
3. Van de Plas, R., Ojeda, F., Dewil, M., Van Den Bosch, L., De Moor, B., Waelkens, E.: Prospective exploration of biochemical tissue composition via imaging mass spectrometry guided by principal component analysis. In: Proceedings of the Pacific Symposium on Biocomputing, Maui, vol. 12, pp. 458–469 (2007)
4. McCombie, G., Staab, D., Stoeckli, M., Knochenmuss, R.: Spatial and spectral correlations in MALDI mass spectrometry images by clustering and multivariate analysis. Analytical Chemistry (19), 6118–6124 (2005)
5. Hanselmann, M., Köthe, U., Kirchner, M., Renard, B.Y., Amstalden, E.R., Glunde, K., Heeren, R.M.A., Hamprecht, F.A.: Toward digital staining using imaging mass spectrometry and random forests. Journal of Proteome Research 8(7), 3558–3567 (2009)
6. Luts, J., Ojeda, F., Van de Plas, R., De Moor, B., Van Huffel, S., Suykens, J.A.K.: A tutorial on support vector machine-based methods for classification problems in chemometrics. Analytica Chimica Acta 665(2), 129–145 (2010)
7. Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., Knight, K.: Sparsity and smoothness via the fused lasso. Journal of The Royal Statistical Society Series B 67(1), 91–108 (2005)
8. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. The Annals of Statistics 32(2), 407–451 (2004)
9. Chung, F.R.K.: Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, vol. 92. American Mathematical Society, Providence (February 1997)
10. Li, C., Li, H.: Network-constrained regularization and variable selection for analysis of genomic data. Bioinformatics 24(9), 1175–1182 (2008)
11. Signoretto, M., Daemen, A., Savorgnan, C., Suykens, J.A.K.: Variable selection and grouping with multiple graph priors. In: 2nd Neural Information Processing Systmes (NIPS) Workshop on Optimization for Machine Learning (2009)
12. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society, Series B 67, 301–320 (2005)
13. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research 7, 2399–2434 (2006)
14. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society, Series B 68, 49–67 (2006)
15. Van de Plas, R., Pelckmans, K., De Moor, B., Waelkens, E.: Spatial querying of imaging mass spectrometry data: A nonnegative least squares approach. In: Neural Information Processing Systems Workshop on Machine Learning in Computational Biology (2007)

Part V

Molecular Structure Prediction

A Matrix Algorithm for RNA Secondary Structure Prediction

S.P.T. Krishnan[1], Mushfique Junayed Khurshid[2], and Bharadwaj Veeravalli[2]

[1] Institute for Infocomm Research,
Fusionopolis, Connexis South Tower, #21-01
1 Fusionopolis Way, Singapore 138632
[2] National University of Singapore,
21 Lower Kent Ridge Road, Singapore 119077
krishnan@i2r.a-star.edu.sg, {mushfiquejk,elebv}@nus.edu.sg

Abstract. In this paper we propose a novel high-performance algorithm, referred to as MARSs (Matrix Algorithm for RNA Secondary Structure Prediction), for predicting RNA Secondary Structures with or without pseudoknots. The algorithm is capable of operating in both serial and parallel modes. The algorithm will take complete advantage of the explicit hardware parallelism increasingly available in todayś multi-core processors resulting in execution speedups. Unlike Dynamic Programming based algorithms, MARSs is non-recursive by design and therefore eliminates some of the disadvantages of Dynamic Programming based algorithms. We performed a large-scale experiment on a multi-core hardware using real sequences with verified structures. We detail and discuss the results from these experiments using metrics such as performance gains, run-times and prediction accuracy. This is one of the first attempts of its kind to provide a complete flexibility in evolving a RNA secondary structure with or without pseudoknots using a matrix-based approach.

Keywords: RNA secondary structure prediction, parallel computing, high performance computing, multi-core.

1 Introduction

RNA plays several important roles in a living cell – carries genetic information, acts as catalyst for various cellular processes and also plays a vital role in gene expression. RNA can be represented in one of three forms - primary, secondary and tertiary. The tertiary structure of RNA determines its function and the RNAs *secondary structure* significantly affects the three dimensional shape of the RNA. Therefore, predicting RNA secondary structures is key in determining the 3D structure of a RNA molecule and also in inferencing the RNAś functions and behaviors.

The research community have over several years proposed several RNA secondary strucure prediction algorithms. Many current & popular RNA secondary structure prediction algorithms are based on Dynamic Programming derivatives.

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 337–348, 2010.

Some of the well-known algorithms are as follows - Nussinov's algorithm[1], Waterman's algorithm [2], MFOLD [3], PKNOTS [4], Akutsu's algorithm [5], Dirks' algorithm [6], PKNOTSRG [7], Jitender's algorithm [8] and Ruan's algorithm [9]. Many Dynamic Programming based algorithms are unable to predict pseudoknots and restrict themselves to predicting secondary structures comprising of only simple structural motifs.

In the last 2 decades, the computing industry has been following moore's law and making faster chips year-on-year basis. Traditionally, the speed gain was obtained by increasing the clock speed of the processor core. However, during the last few years the processor manufacturers have been adding more processing cores instead of making a single core faster due to physical sciences limitations. This requires redesign of current algorithms in many industries. Dynamic Programming algorithms are recursive by design and this inhibits the algorithm from being parallelized on a multi-core system. The primary road block being that later iterations depend on the results from earlier iterations. This requires the design and development of a new class of algorithm that works equal well on single-core and multi-core architectures.

In this paper, we introduce a new algorithm for predicting RNA secondary structures called *MARSs* (Matrix Algorithm for RNA Secondary Structure prediction). MARSs does not use the popular dynamic programming methods and therefore we believe it to be the first-of-its-kind. MARSs is capable of predicting both Pseudoknots and Non-Pseudoknot structures with equal ease. MARSs has been designed with parallelism in mind and can easily scale from single-core to many-cores resulting in significant speedup and with no degradation in the quality of output. We used MARSs to predict secondary structures of real RNA sequences and also observed the speedup as a result of using many incremental multi-cores system with each subsequent system have one more processing core than the previous system. We report these results in this paper.

2 RNA Secondary Structures

RNAs are functionally important and play a key role in various cellular processes such as ribosomal frame-shifting, control of translation and splicing [10]. RNA primary structure is the sequence of nucleotide bases that comprise the RNA, namely **A** (adenine), **C** (cytosine), **G** (guanine) and **U** (uracil). Base pairings of these nucleotides are the main determinants of the secondary structures of RNA. Stable base pairs in RNA like *Watson-Crick* (**A-U** and **G-C**) and *Wobble* (**G-U**) are common, but weaker base pairs like *Hogsteen* (**A-C**) are also possible. Moreover, there are some common structural motifs that occur in RNA Secondary structures: Double-stranded segment, Bulges, Symmetric or Asymmetric internal loops, Hairpins, Two-stem junctions (coaxial stacks), Kissing hairpins [11].

In addition to the above simple motifs, *Pseudoknots* are also a common occurrences in RNA. *Pseudoknots* are an important structural element present in RNA secondary structures such as Ribozymes [14]. During 1980s *Pseudoknots* were described to be a section of structures in virus RNAs in plants, which resemble tRNAs. Several important biological processes rely on RNA molecules with pseudoknots. For example, the RNA component of human telomerase contains a pseudoknot that is critical for activity [16]. The structures formed by pseudoknots are also crucial building block of RNA tertiary structure. Hence it is of importance to predict not only simple motifs but also *Pseudoknots.*

Pseudoknots can be defined as follows: If R is an RNA sequence such that $R = r_1, r_2, r_3, r_4 ... r_n$, and $(r_x r_y)$ and $(r_p r_q)$ are two different base pairs existing in this RNA structure ($x < y$ and $p < q$), a pseudoknot is composed of these two base pairs when $1 \leq x < p < y < q \leq n$ [13]. Simply stated, a *Pseudoknot* is a nucleic acid secondary structure containing at least two stem-loop structures in which a section of one stem is intercalated between the two sections of another stem. Pseudoknots fold into knot-shaped three-dimensional conformations but are not true topological knots.

3 MARSs

In this section, we will describe our MARSs algorithm from a design perspective and contrast it with Dynamic Programming algorithms used for RNA secondary structure prediction.

3.1 Overview

MARSs uses a top-down prediction methodology unlike Dynamic Programming algorithms that employ a bottom-up approach. In a top-down approach motifs are generated on a global scale and then the local regions are explored. MARSs stores all the intermediate results, as with Dynamic Programming algorithms, in order to prevent duplication of effort. Unlike Dynamic Programming based algorithms MARSs does not view the secondary structure prediction as a set of overlapping sub-problems. This key distinction makes MARSs high-scalable and easily portable to architectures that have large number of parallel processing units.

Given a primary RNA structure, MARSs produces several possible RNA secondary structures. This is also a key difference between MARSs and Dynamic Programming based algorithms where the latter typically produces a single secondary structure. MARSs has several tuning parameters to its base-pair maximization engine that can be adjusted to produce varying number of output structures. A bioinformatician may use these to inject knowledge from wet-lab experiments and impact the behavior of MARSs algorithm. We believe this attribute makes MARSs algorithm future-proof.

MARSs algorithm does not employ a dictionary-based approach in identifying local structural motifs. Yet, it can predict all known RNA structural motifs and by extension is capable of finding newer types of motifs - either composite of known motifs or entirely newer ones.

MARSs is engineered to be parallelized and consists of several independent processes. Therefore, unlike Dynamic Programming based algorithms, MARSs can be easily parallelized and significant speed up can be achieved when executed on high performance computers with multiple cores. MARSs consistently shows a speed up of close to N, where N is the number of cores.

MARSs also shows high levels of prediction accuracies. In our experiments comprising of 100 real sequences with verified 2^o structures, MARSs produced an average PPV (Positive Predicted Value) of **76.46%** and the average sensitivity was **81.04%**. Hence, MARSs is ready to be used with RNA sequences with unknown 2^o structures. These PPV and Sensitivity values compares well with other State-of-the-art algorithms (see results section).

3.2 Matrices and Folding

The foundation of the MARSs algorithm is basically built upon two matrices: namely the **Base Pair matrix (or BP matrix)** and the **Affinity Matrix (or AM)**. The BP matrix remains static throughout the running of the algorithm. It basically represents the base-pairing affinities between all possible nucleotides. It is a 4x4 matrix with the row and column representing known RNA alphabets. The base pair matrix used in our experiments is shown in Figure 1.

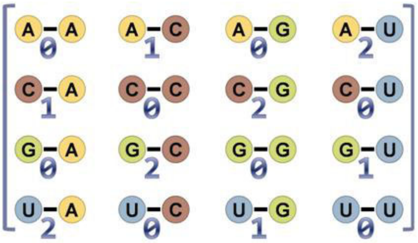


Fig. 1. Base Pair Matrix

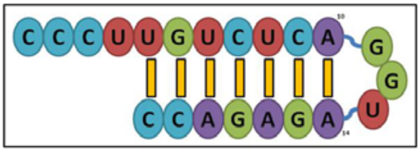


Fig. 2. Folding across 10 and 14 and attempted bonds

The score of each of the possible base pairs in the BP matrix are assigned using general base pairing rules like *Watson-Crick, Hogsteen, Wobble* etc. In the above case, *Watson-Crick* pairs (**G-C** and **A-U**) are given a score of 2 while *Hogsteen* (**A-C**) and *Wobble* (**G-U**) base pairs are given a score of 1. A base pair in the BP matrix has a score of 0 when the base-pairing is not possible. The table can be updated with precise values of bond probabilities between the base pairs and a re-run of MARSs will predict the newer structures.

The given RNA primary structure, i.e. the nucleotide sequence, and this static BP matrix is what is used to initially construct the AM. This AM is initialised during runtime. For a given sequence of nucleotides, each nucleotide, from one end to the other, are numbered from 0 to $N-1$, where N is the length of the nucleotide sequence. This numbering is like array notation in computer programming languages so we can refer to any specific nucleotide in the sequence using integers 0 to $N-1$. Now we proceed to construct an NxN matrix (which is the AM), the row and column numbers of which refer to the nucleotide numbers as mentioned. Each element at co-ordinates (x, y) of the AM represents the score of a possible base pairing between nucleotide number x and nucleotide number y. This matrix is filled by referring to the BP matrix.

In our experiments, we used real pseudoknot structures retrieved from the pseudoknot database called *Pseudobase*. Let us take, for example, a pseudoknot present in the RNA of brome mosaic virus, having a PKB-number of **PKB155** in *Pseudobase* [14].

The AM consists of all possible interactions between nucleotides in the RNA sequence. Hence we hypothesize that all the possible secondary structures are present in this master matrix. By crawling this matrix in specific pre-determined ways we can extract the correct structures of this sequence at much less computational cost than dynamic programming. The RNA consists of nucleotides which are strung together naturally in a chain. Hence base-pairing is not possible between two consecutive nucleotides in the sequence. Thus the next step in the algorithm is to zero all the consecutive nucleotides' interactions in the AM, which we call the Neighbor pair rule. Mathematically, they are elements of co-ordinates $(i, i + 1)$ and $(i - 1, i)$, where $0 \leq i \leq N - 1$.

As the RNA secondary structure is formed by the primary sequence folding upon itself, the next step we do in our algorithm, is to fold the RNA sequence in various ways. This initial folding can give rise to hair pin loops. The folding can occur across any pair of nucleotides in the RNA sequence. Since all nucleotides can pair with each other, the number of possible folds are: $N(N - 1)/2$. In the AM, each element can also be considered to be a folding point. For example, figure 2 shows the shape of the RNA after we just fold it across bases number 10 and 14, including all the attempted bonds (explained later).

3.3 Base Pairing - Symmetric Fold

The next step is to form base pairs in the exposed local regions using the first fold as the anchor. We have added two algorithms for this purpose. The algorithms produce secondary structure with symmetrical and asymmetrical structural motifs. In this way, we are able to generate structures that consist of all types of known structural motifs.

The first of the two base-pairing algorithm is called **S-fold (or Symmetric Fold)** base pairing. For each of the folds, we try and form bonds along bases that are directly facing each other after the fold. For example, in Figure 2, the bases along which the fold has been made are numbered (in this case 10 and 14), and the *orange* lines are the base-pairings that our S-fold base pairing algorithm will try to form. The bases at both ends of the primary sequence that never form bonds with any other base, contribute to form potential dangling ends. In this case these bases are the ones with no orange line attached to it. When no bonds can be formed between opposing nucleotides and are surrounded by base pairs, internal loops are born. To form a base pair, we refer to the AM element corresponding to those two nucleotides, and if it is a value other than 0, we add the score to the score of that structure and form a base pair. Once a base pair is formed, we also assign 0 to the entire corresponding rows and columns in the AM of the nucleotides involved in the base pair. This is done as these two nucleotides cannot bond with any other nucleotides in this particular fold. This process effectively means we are traversing the AM along a 45 degree line, starting from the folding point and forming bonds whenever the element has a value of greater than 0. An AM numerical representation of the folding starting from nucleotides 10 and 14 is shown in Figure 3. The *dark green* shaded element is the folding point, and the *blue* elements are the base pairs that are formed in case of S-fold. The *grey* area is the hair-pin loop. The *light brown* elements are represents the dangling ends. The *red*, *orange* and *yellow* colored elements represent strong, weak & no bonds and are not used in this level 1 secondary structure.

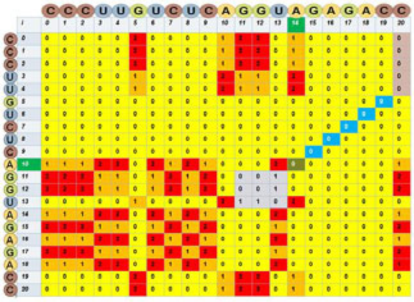

C C C U U G U C U C A G G U A G A G A C C
i 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Fig. 3. AM representation of folding and S-fold bonding across bases 10 and 14

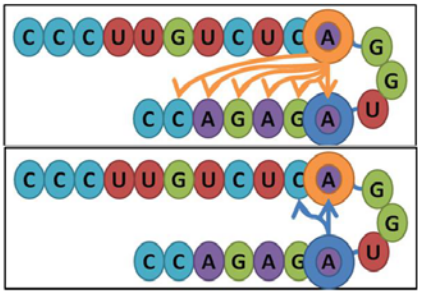

C C C U U G U C U C A G G U A G A G A C C
C C C U U G U C U C A G G U A G A G A C C

Fig. 4. Base Pairs tried in A-fold

Using this method, multiple anchor folds are formed and for every anchor fold the possible base-pairs are formed. At the end of the process, we have a set of structures that we refer to as level 1 secondary structures. These structures do not have pseudoknots yet, but have hair-pin loops, stems, internal loops and possible dangling ends.

3.4 Base Pairing - Asymmetric Fold

The second scheme, Base Pairing - Asymmetric Fold scheme, can predict all types of level 1 structural motifs that have some sort of asymmetry about them, like bulges and asymmetric internal loops. Once the anchor fold is determined, this base pairing algorithm using two pointers begins traversal along the two *arms* about the fold. As an example, suppose a fold is determined by the folding algorithm to be at nucleotides (say 10 and 14), then this A-fold base pairing would place two pointers at the two folding nucleotides, as shown in Figure 4.

So in this example, let us assume that the fold is made across the *orange* and *blue* circled bases. The circles represent the pointers, and hence the two pointers are based at these two nucleotides. Now these two pointers try forming base pairs in the following manner -

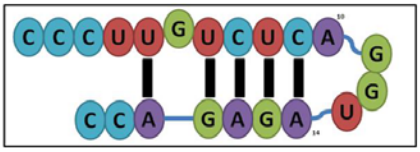


Fig. 5. Level 1 structure after A-fold base pairing across 10 and 14 of PKB155

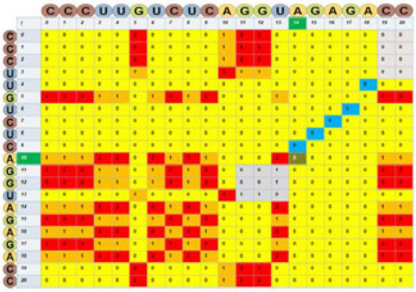


Fig. 6. AM representation of Level 1 A-fold structure

1. The pointer tries to make a base pair with the base of the opposite pointer
2. If the AM does not support such a base pair, it moves one base along the opposite arm and tries again.
3. If AM still does not support such a base pair, it again moves one base, and keeps moving till a pre-defined limit is reached. In our case we placed a limit of 6 nucleotides, since we observed a maximum bulge of 6 nucleotides in *Pseudobase* [14]. If limit is reached, then it does not form any base pair.
4. Both the pointers follow the above steps. Figure 4 shows the base pairs that are attempted initially by the pointers in our hypothetical example.

The blue pointer tried only 2 base pairs, since the second one forms a base pair (**A-C**, *Watson-Crick*) and hence the pointer stopped traversing. While the orange pointer had to traverse to the limit (6 nucleotides) until it could find a possible base pair (**A-C**, *Watson-Crick*). So, following the above procedure, each of the two pointers will produce none or one base pair. Therefore, we need a protocol or precedence to decide which base pair to choose if both pointers each form one base pair? We only need to make a choice in the scenario (common) that both pointers yield a different base pair. We follow the criteria below in order of preference in order to decide the final base pair.

1. Choose the base pair that had to skip least number of nucleotides to form the base pair. We call this *distance*.
2. If *distance* is same, then we pick the base pair which has more weight, ie. a *Watson-Crick* base pair is chosen over a *Hogsteen* base pair.
3. If both the base pairs' weight are the same, we arbitrarily choose one pointer.

If a base pair is formed, the two pointers each then move to the nucleotide next to the base-pair just formed and if no base pair is formed the pointers move one base away from its previous position. Then the whole process is repeated until one of the pointers reach the end of the strand. At the end of this process, a level 1 structure is produced by A-folding. Figure 5 shows the level 1 structure produced by A-folding for our hypothetical sequence with the anchor fold at nucleotides across 10 and 14. We can now observe that a bulge is formed, hence asymmetry is introduced in this algorithm. Figure 6 shows the AM representation of this structure. We can see that the traversal is no longer 45 degrees from the folding point, but made of broken lines, to introduce asymmetry.

3.5 Level 2 Folding

Now the next step is to form Level 2 secondary structural elements, which is Level 2 folding. We currently have generated N^2 different level 1 structures corresponding to folds. As mentioned earlier, for each of these structures we change the AM by marking the rows and columns corresponding to the nucleotides involved in base-pairs in the Level 1 structure to be 0, since they can no longer interact with other nucleotides. So now, in order to form Level 2 secondary structural elements, we *fold* the Level 1 structures again and run S-fold base-pairing for each level 1 fold to find more base pairs that can cause pseudoknots or coaxial stacks. A level 2 folding may or may not result in a pseudoknot, as shown in Figure 7.

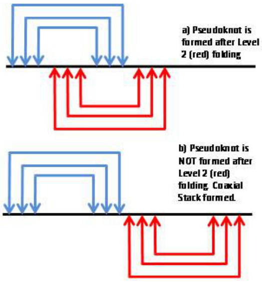

a) Pseudoknot is formed after Level 2 (red) folding
b) Pseudoknot is NOT formed after Level 2 (red) folding. Coaxial Stack formed.

Fig. 7. Illustration of pseudoknots or coaxial stacks

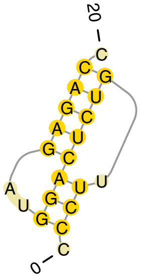


Fig. 8. One predicted structure of PKB155

Let us go back to our test case of Level 1 structure with folding across 10 and 14. After Level 2 folding, one of the high energy (16) (and therefore more likely secondary structure) is as shown in Figure 8. This figure shows Level 2 (level 2 folding across 4 and 9) structures and the corresponding Level 1 [10, 14] structure. All bonds are correctly predicted when compared to the structure in Pseudobase [14], hence the predicted structure has a base-pair distance of 0. It has a PPV of 100% and a sensitivity of 100%.

4 MARSs Complexity Analysis

In this section, we attempt to derive the time and space complexities of MARSs algorithm. It should however be noted that the actual space and time complexities depends on the composition of the nucleotides themselves that in turn affects the set of possible structures. The resource complexities also depends on the tuning parameters that dictate the final number of secondary structures that are desired.

$$\text{Number of folds} = \frac{N(N-1)}{2}$$

Hence, Space complexity is $O(n^2)$

$$\text{Maximum number of potential base pairs traversed in S-Fold} = \frac{N}{2}$$

$$\text{Minimum number of potential base pairs traversed in S-fold} = 1$$

$$\text{Average potential base pairs traversed} = \frac{N+2}{4}$$

$$\text{TIME COMPLEXITY} = \frac{N(N-1)}{2} \times \frac{N+2}{4} = O(n^3)$$

Since A-fold base pairing simply increases the number of base pairs traversed by a constant, hence complexity remains same.

5 Results

5.1 Accuracy Measures

As mentioned before, the accuracy measures that we use to analyze the accuracy of a predicted structure when compared to experimentally verified structures are **Sensitivity** and **PPV**. **Positive Predictive Value (PPV)** is the number of correctly predicted base pairs as a percentage of the total number of base pairs in the predicted structure. Its primary focus is on the accuracy of predicted base pairs, without regard to any unpredicted base pairs. **Sensitivity** is the number of correctly predicted base pairs as a percentage of the total number of base pairs in the experimentally verified structure. Its primary focus is on predicting base pairs present in the actual structure, without regards to the number of false base pair predictions. These two measures are now the standards for measuring accuracy in case of RNA secondary structure prediction [15].

$$\text{PPV} = \frac{\text{number of correctly predicted base pairs}}{\text{total number of base pairs in PREDICTED STRUCTURE}} \times 100\%$$

$$\text{Sensitivity} = \frac{\text{number of correctly predicted base pairs}}{\text{total number of base pairs in ACTUAL STRUCTURE}} \times 100\%$$

A structure can be said to be perfectly predicted, when both the PPV and sensitivity values are 100%. PPV and sensitivity shows the measure of accurate base pairs predictions relative to the predicted and the actual structure respectively.

5.2 Accuracy of MARSs

For our analysis, we selected a set of 100 sequences from *Pseudobase* [14] from different classes as classified in the website. The average PPV of all these sequences is **76.46%** and the average sensitivity is **81.04%** and the average normalized b-p distance is **12.58%**.

It can seen that the average PPV and sensitivity of the MARSs algorithm on the randomly selected sequences is pretty good. In order to be comprehensive, we also compare the predictions with a dynamic programming based RNA secondary structure algorithm and observe the results. We ran the same sequences on *Ruan's server* (which predicts Pseudoknots) [9] that employs the Ruan's algorithm as previously mentioned and also in the software *PKNOTSRG* (as previously described) [7]. Ruan's algorithm showed an average PPV and sensitivity of **56.8%** and **58.9%** respectively, while PKNOTSRG had an average PPV and sensitivity of **70.5%** and **67.8%** respectively. Hence we can see MARSs produced significantly better results.

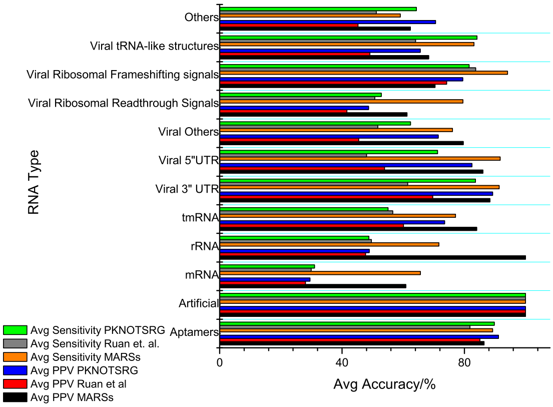


Fig. 9. Ruan et al. and PKNOTSRG compared with MARSs

Figure 9 shows a bar chart with Ruan's and PKNOTSRG's accuracy parameters compared with MARSs, for RNA of different classes as classified by *Pseudobase* [14]. We can see that MARSs appears to be consistently more accurate than Ruan's algorithm and PKNOTSRG in terms of all classes of RNA, except for a few instances like Aptamers, Viral 3' UTR and Viral Ribosomal Frame-shifting signals. Even in those cases it is only slightly less accurate than the other algorithms only and the difference is a very small amount.

6 Parallelization of MARSs

One of the primary design objectives of MARSs is to be easily parallelized and take advantage of the explicit hardware parallelism available on the hosts. This is achieved as follows. After the first Level 1 folding, we obtain a series of Level

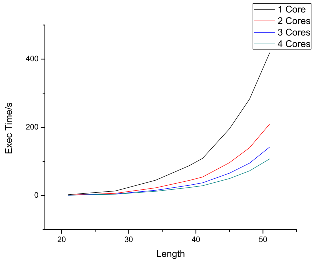


Fig. 10. Multi-core implementation of MARSs in Intel Quad Core Xeon 2992.006 MHz

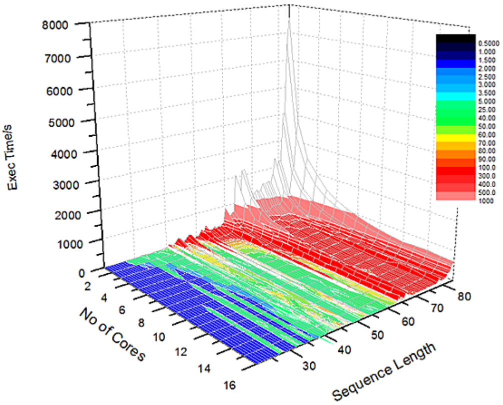


Fig. 11. Colour Mapped Surface for execution times

1 structures, each having its own Affinity Matrices. Hence, when we do Level 2 folding of these structures, they are completely independent of each other. Therefore, we can run each of the Level 2 folding activity on different cores concurrently. The data plot in figure 10 demonstrates how the algorithm speeds up when we run it on multiple cores, while processing different sequence lengths. For the plot, the algorithm was run in Intel Quad Core Xeon machine with a clock speed of 2992.006 MHz.

From figure 10, we can clearly see that as the number of cores increased, the execution time significantly decreases. The entire set of 100 sequences from *Pseudobase* [14] was then run on a second machine. This machine is a virtual machine having 16 virtual cores being run by the software QEMU version 0.9.1. This virtual machine is actually powered by 12 physical cores, each being Intel Xeon CPU E7450 2.40 GHz. The graph in figure 11 shows a colour mapped surface plot of execution time vs number of cores vs sequence length. We assigned more colour codes for lower times, since the trend is less apparent for the subtle change in shape of the graph. We can see the clear trend in decreasing execution time as number of cores are increased due to parallelization, and also we can see for smaller sequences, too high number of cores tend to slow it down. Also, for large sequence lengths, we can see the enormous significance of decrease in run time, as number of cores are increased.

7 Conclusion

In this paper, we have introduced a new high-performance and scalable algorithm for RNA secondary structure prediction. The algorithm is designed to be parallelized and strives to be fundamentally different when compared to the generic class of RNA secondary structure prediction algorithms that are Dynamic Programming based, thus pushing the state-of-the-art to the next level. The algorithm is shown to take complete advantage of the explicit hardware parallelism available with our experiments. We have conducted experiments to quantify the performance of the algorithm. For this, we extracted RNA sequences with experimentally verified structures from Pseudobase and compared MARSs predictions with actual structures. We have also conducted a 2^{nd} experiment with a focus on performance improvement through parallelization. For this, we employed a 16 core Intel 64-bit server. The results indicate that the algorithm produces much better results when compared with other algorithms and also is scalable to use hardware parallelization features. The algorithm however is still in its early stages and extensive experiments are to be conducted on actual very large-scale structures to identify its limitations, if any. This would be an immediate extension to the problem under study.

References

1. Nussinov, R., Piecznik, G., Grigg, J.R., Kleitman, D.J.: Algorithms for loop matchings. SIAM Journal on Applied Mathematics 35, 68–82 (1978)
2. Waterman, M.S., Smith, T.F.: Rapid dynamic programming methods for RNA secondary structure. Adv. Appl. Math. 7, 455–464 (1986)
3. Lyngs, R., Zuker, M., Pedersen, C.: An Improved Algorithm for RNA Secondary Structure Prediction. Tech-report BRICS RS-99-15 (1999)
4. Rivas, E., Eddy, S.: A dynamic programming algorithm for RNA structure prediction including pseudoknots. J. Mol. Biol. 285(5), 2053–2068 (1999)
5. Akutsu, T.: Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. Discrete Applied Mathematics 104, 45–62 (2000)
6. Dirks, R., Pierce, N.A.: A partition function algorithm for nucleic acid secondary structure including pseudoknots. Journal of Computational Chemistry 2003 24, 1664–1677 (2003)
7. Reeder, J., Giegerich, R.: Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. BMC Bioinformatics 5, 104 (2004)
8. Deogun, J., Donts, R., Komina, O., Ma, F.: RNA Secondary Structure Prediction with Simple Pseudoknots. In: APBC 2004, pp. 239–246 (2004)
9. Ruan, J., Stormo, G.D., Zhang, W.: ILM: a web server for predicting RNA secondary structures with pseudoknots. Nucleic Acids Research 32(Web Server Issue), W146–W149 (2004)
10. Ren, J., Rastegari, B., Condon, A., Hoos, H.H.: HotKnots: Heuristic prediction of RNA secondary structures including pseudoknots. RNA 11, 1494–1504 (2005)
11. Tian, B., Bevilacqua, P.C., Diegelman-Parente, A., Mathews, M.B.: The double-stranded-RNA-binding motif: interference and much more. Nature Reviews Molecular Cell Biology 5, 1013–1023 (2004)
12. Brion, P., Westhof, E.: Hierarchy and dynamics of RNA folding. Annu. Rev. Biophys. Biomol. Struct. 26, 113–137 (1997)
13. Fu, X.Z., Wang, H., Harrison, W., Harrison, R.: RNA Pseudoknot Prediction Using Term Rewriting. International Journal of Data Mining and Bioinformatics (2006)
14. Batenburg, F.H., Gultyaev, A.P., Pleij, C.W.: PseudoBase: structural information on RNA pseudoknots. Nucleic Acids Research 29(1), 194–195 (2001)
15. Ding, Y., Chan, C.Y., Lawrence, C.E.: RNA secondary structure prediction by centroids in a Boltzmann weighted ensemble. RNA 11, 1157–1166 (2005)
16. Chen, J.L., Greider, C.W.: Functional analysis of the pseudoknot structure in human telomerase RNA. Proc. Natl. Acad. Sci. USA 102(23), 8080–8085 (2005)

Exploiting Long-Range Dependencies in Protein β-Sheet Secondary Structure Prediction

Yizhao Ni and Mahesan Niranjan

ISIS Group, School of Electronics and Computer Science
University of Southampton, U.K
Yizhao.NI@googlemail.com,
mn@ecs.soton.ac.uk

Abstract. We investigate if interactions of longer range than typically considered in local protein secondary structure prediction methods can be captured in a simple machine learning framework to improve the prediction of β sheets. We use support vector machines and recursive feature elimination to show that the small signals available in long range interactions can indeed be exploited. The improvement is small but statistically significant on the benchmark datasets we used. We also show that feature selection within a long window and over amino acids at specific positions typically selects amino acids that are shown to be more relevant in the initiation and termination of β-sheet formation.

Keywords: Protein Secondary Structures, β-Sheet, Feature Selection, Machine Learning.

1 Introduction

Predicting the secondary structure of proteins from their amino acid sequences using machine learning methods has been of interest for several decades. Examples of early work in the topic include that of Qian and Sejnowski [12]. Work in the area appears to have stabilized over the years, with the availability of several stable web based prediction servers (e.g. JPred [2] and its previous incarnations). An overview of development in the area approximately halfway through the period of the above papers is given by Rost [13].

The basic strategy for prediction of secondary structure has largely been to encode a local window of amino acids (usually $11-15$), using a one in Ω binary coding method, where $|\Omega| = 20$, leading to an input space of dimension in the range $220-300$. The output space is usually three dimensional predicting if the secondary structure at the centre of the window (namely the *central residue*) is an α-helix, β-sheet or of an unspecified structure, usually referred to as coil. A mapping between such a multivariate input and the three dimensional output space can be learned by a machine learning technique of one's choice, in which artificial neural networks of the multi-layer perceptron type [11] is the most popular in the literature.

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 349–357, 2010.

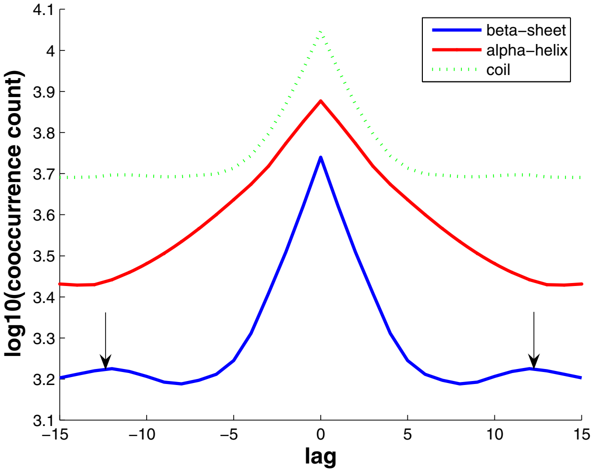


Fig. 1. Distribution of co-occurrences of secondary structures separated by a lag from the central residue of the input window for the three secondary structure classes. Arrows show that for the β-sheet there is some long range interaction outside the usually considered analysis lengths.

Of the three classes usually considered for predictions in this setting, it is known that β-sheets are the most difficult to predict. This observation is usually attributed to the fact that sheet structures are formed by interactions of longer range than is accommodated within the local windows. The obvious solution to dealing with this by increasing the window length is usually not expected to be successful because with each additional position included, we increase the dimensionality by 20, and a corresponding increase in the amount of training data will be required.

In this paper we explore the possibility of longer windows for β-sheet prediction with feature subset selection to keep the input dimensionality low. We first observe, using co-occurrence counts, that β-sheets contain a small amount of long range dependencies. Fig. 1 shows this co-occurrence counts for the three classes of secondary structures, where we plot the counts at different position lags from the central residue to a logarithmic scale. We observe a small but noticeable difference between the β-sheet and the other two classes. Motivated by this observation, we show that *recursive feature elimination* picks up a small subset of amino acids and their positions in the window to achieve a quantifiable improvement in prediction accuracies.

2 Materials and Methods

2.1 Kernel Classifiers

Let us denote the protein sample pool as $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m$, where $\mathbf{x}_i = (x_i^1, x_i^2, \ldots, x_i^{N_i})$ is the i-th amino acid sequence with N_i denoting the length of the sequence, $x_i^j \in \Omega$ and Ω represents the set of amino acids appeared in $\mathcal{S}$. Similarly, the i-th secondary structure sequence is denoted by $\mathbf{y} = (y_i^1, y_i^2, \ldots, y_i^{N_i})$ where $y_i^j \in \{1, -1\}$ with 1 representing β-sheet (E) and -1 otherwise (∼E). Whenever this can be done without loss of clarity, each example (x_i^j, y_i^j) is also abbreviated as (x_n, y_n), where the number of examples is defined by $N = \sum_{i=1}^{m} N_i$.

In order to solve the presented binary classification problem (i.e. E and ∼E), the support vector machine (SVM) technique is applied. It learns a linear operator $\mathbf{w}$ by solving the following optimisation problem

$$\begin{array}{ll} \min_{\mathbf{w}, w_0, \boldsymbol{\xi}} & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\mathbf{1}^T\boldsymbol{\xi} \\ s.t. & y_n\big(\mathbf{w}^T\phi(x_n) + w_0\big) \geq 1 - \xi_n \quad n = 1, \ldots, N \\ & \boldsymbol{\xi} := \{\xi_n | \xi_n \geq 0, \, n = 1, \ldots, N\} \end{array} \tag{1}$$

such that a new amino acid residue x has the prediction $f(x) = sgn(\mathbf{w}^T\phi(x))$, where $\phi(x) \in \mathbb{R}^D$ is an embedding feature function which will be specified in Section 2.2 and $sgn(\cdot)$ indicates the sign of the expression.

In addition, one can turn to solving the dual representation of (1)

$$\begin{array}{ll} \max_{\boldsymbol{\alpha}} & -\frac{1}{2}\boldsymbol{\alpha}^T\mathbf{K_{xy}}\boldsymbol{\alpha} + \mathbf{1}^T\boldsymbol{\alpha} \\ s.t. & \mathbf{y}^T\boldsymbol{\alpha} = 0 \\ & \boldsymbol{\alpha} = \{\alpha_n | 0 \leq \alpha_n \leq C, \, n = 1, \ldots, N\} \end{array} \tag{2}$$

which allows the use of kernels

$$\mathbf{K_{xy}} = \{y_k y_l \phi(x_k)^T \phi(x_l) : \; k, l = 1, \ldots, N\}, \tag{3}$$

and it is expected to provide more flexibility in the feature expression.

2.2 Feature Extraction

Following [6,9], we consider the *position-dependent residue* features extracted from the amino acid sequences. Mathematically, the feature expression is given by the formula

$$\phi_u^p(x_n) = \delta(x_{n+p}, u), \tag{4}$$

with the indicator function $\delta(\cdot, \cdot)$, $u \in \Omega$ and $p = \{-d_l, \ldots, d_r\}$. Fig. 2 illustrates an example. To predict the secondary structure of the n-th central residue, a windowed residue environment $(x^{n-d_l}, \ldots, x^{n+d_r})$ is selected, from which the position-dependent residue features are extracted. As discussed in [6], a proper window size can lead to good performance, because a too short residue segment (e.g. the green box in Fig. 2) may omit some important classification information while a too long segment (e.g. the red box in Fig. 2) may decrease signal-to-noise ratio. Although a reasonable window size (e.g. the blue box in Fig. 2) seems to be a perfect fit, as pointed out in [15], the β-sheets are formed between two strings of complementary residues that maybe distantly separated in the protein sequence, and a long segment is probably beneficial in β-sheet classification. This

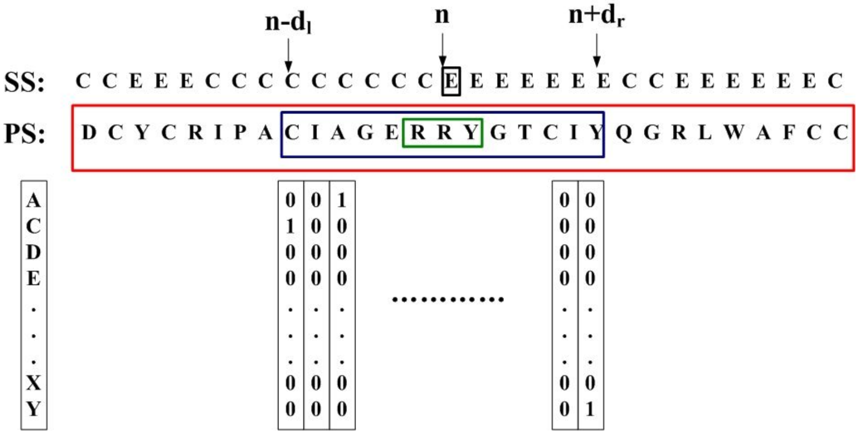

n-d_l
n
n+d_r
SS: C C E E E C C C C C C C C C E E E E E E E C C E E E E E E C
PS: D C Y C R I P A C I A G E R R Y G T C I Y Q G R L W A F C C
A C D E . . . X Y
0 1 0 0 . . . 0 0
0 0 0 0 . . . 0 0
1 0 0 0 . . . 0 0
....
0 0 0 0 . . . 0 0
0 0 0 0 . . . 0 1

Fig. 2. Schematic diagram of encoding a window of amino acids to predict the secondary structure at the centre position (i.e. the central residue). PS and SS denote the primary sequence and secondary structure labels respectively. A local windowed residue environment $(x^{n-d_l}, \ldots, x^{n+d_r})$ is defined and the presence of each amino acid is encoded using a one out of Ω binary coding scheme as shown. These vectors are concatenated to form the high dimensional input space from which predictions are made via a classification method.

poses a dilemma for current research on predicting the secondary structure of proteins (particularly on β-sheet classification), and more sophisticated machine learning technologies are required. In order to capture long-range dependencies in β-sheet secondary structures and show that they can indeed be exploited, we compare two window size setups in the experiments: one is length-13 (i.e. $d_l = 6$ and $d_r = 6$) that is commonly used [6,9]; the other is length-31 (i.e. $d_l = 15$ and $d_r = 15$), with the intention of exploiting long-range interactions. For the datasets we used, there are 286 features for the length-13 setup; by extending the window size to length-31, the dimensionality of feature space increases to 682, leaving the classifier a feature exploitation challenge.

2.3 Data Sets and Experiment Setup

Two sets of non-homologous protein chains, namely RS126[1] and CB513[2], are studied in the experiments, where the automatic assignments of secondary structure to experimentally determined 3D structures are performed by DSSP [7].

[1] The set of 126 non-homologous globular protein chains is used in [14] and has been tested by many current secondary structure prediction methods. It contains 23,349 residues with 32% α-helix, 23% β-sheet, and 45% coil. Therefore, when treated as a binary-class classification problem, the data set contains few positive examples.

[2] The set of 513 protein sequences was constructed by [3], which includes almost all the sequences in the RS126 dataset. It contains 84,119 residues of which 22.7% are β-sheets.

Different from to [6,9,15], we reduced the eight classes of the DSSP assignments to a binary state: E (β-sheet) and B (β-bridge) to E, and all other states to ∼E. A seven fold cross validation[3] was then carried out to estimate the predictive accuracy. Following [15], the statistical significance of differences in prediction quality between window size setups was then evaluated by a paired t-test over the cross-validation results. To avoid the selection of extremely biased partitions, the `RS126` (or `CB513`) dataset was randomly divided into seven subsets with each subset having similar size of each type of secondary structures.

As experienced in the literature, the secondary structure prediction task tends to be non-linear and the *radial basis function* (RBF) kernel is commonly used [6,9,15]. Therefore, we also adopt the RBF kernel

$$K(x_k, x_l) = \exp(-\gamma \|\phi(x_k) - \phi(x_l)\|^2) \tag{5}$$

for optimisation (2), where the parameter γ is tuned by cross-validation.

Finally, *the area under the ROC curve* [1] is applied to evaluate the performance of SVM with different window size setups.

3 Results and Discussion

Tables 1 and 2 show the classification performances of linear and RBF classifiers, and the RBF classifier working with the best selected subset of features on the two datasets used. We first note that increasing window length improves performance, implying that some long-range residue patterns are helpful in detecting β-sheets. This is consistent with the postulation discussed in [15].

The classification performance of SVM with RBF kernels displayed in Table 1 and Table 2 is consistently better than SVM with linear kernels on both data sets. Moreover, in this scenario SVM with length-13 performs better than SVM with length-31, which is consistent with the "concern" in [6]. We believe that this is due to interference terms of irrelevant residue patterns brought in by the long window size. Since the feature space of the RBF kernel is of the form [8]

$$\varphi(x) = \exp(-\gamma\|\phi(x)\|^2)\Big(\sqrt{\frac{(2\gamma)^k C_{\boldsymbol{\theta}}^k}{k!}}\,\phi(x)^{\boldsymbol{\theta}}\Big)_{|\boldsymbol{\theta}|=k,k=0}^{\infty} \tag{6}$$

where $\boldsymbol{\theta} = \big\{(\theta_i)_{i=1}^D | \theta_i \in \mathbb{N}, |\boldsymbol{\theta}| = \theta_1 + \ldots + \theta_D = k\big\}$, $C_{\boldsymbol{\theta}}^k = \frac{k!}{\theta_1! \ldots \theta_D!}$ and $\phi(x)^{\boldsymbol{\theta}} = \phi_1(x)^{\theta_1} \cdots \phi_D(x)^{\theta_D}$; each feature would have influence on many other features. In this case, irrelevant residue patterns can decrease the signal-to-noise ratio severely and deteriorate performance.

In order to reduce or eliminate the influence of irrelevant features (i.e. residue patterns at specific positions), we applied the *Recursive Feature Elimination* (RFE) [5] technique to select important features (RFE-RBF). Specifically, the length-31 features are first ranked by a linear SVM with RFE[4]. To speed up

[3] The seven fold cross validation setup is inherited from [6,9].

[4] We also tried to rank features by a RBF SVM with RFE, however, this setup biased towards very rare features, which conversely destroyed the performance (the results are not shown in this paper).

Table 1. Prediction performances on the RS126 dataset, as measured by areas under ROC curves, at two different window lengths and with feature elimination from the longer of the windows. Performance of linear and RBF kernels are shown. *P*-values of *T*-test for statistical significance in the differences between each method and the RFE-RBF method (results in bold) are shown in the lower part of the table.

Window size	**The area under the ROC curve**	
	LINEAR kernel	RBF kernel
length-13	75.24 ± 1.19	77.30 ± 0.83
length-31	76.22 ± 0.85	76.80 ± 0.75
RFE-RBF	N/A	$\mathbf{77.65} \pm 0.75$

P-value in T-test		
Window size	LINEAR kernel	RBF kernel
length-13	$3.60e-4$	$3.98e-2$
length-31	$1.97e-4$	$3.70e-3$

Table 2. Prediction performances on the CB513 dataset. See caption of Table 1.

Window size	**The area under the ROC curve**	
	LINEAR kernel	RBF kernel
length-13	75.59 ± 0.50	78.28 ± 0.64
length-31	76.96 ± 0.59	78.03 ± 0.73
RFE-RBF	N/A	$\mathbf{78.78} \pm 0.73$

P-value in T-test		
Window size	LINEAR kernel	RBF kernel
length-13	$7.86e-7$	$1.00e-3$
length-31	$1.25e-7$	$4.00e-5$

the process, we eliminate 10 features each time. A proportion of the top ranked features is then selected and the RBF kernel is constructed using these features only. For the experiments on the RS126 dataset, the proportion is taken from $\{\frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}\}$ and the performance with respect to the proportion of features is depicted in Fig. 3. We observed that when the proportion increases, the performance first increases because of the increasing contribution of features to the classification. But after a certain point (i.e. $\frac{1}{2}$ in this experiment), the performance decreases, possibly due to the influence of irrelevant features. In addition, if we choose a proper proportion[5], we are able to obtain better performance compared with SVM with length-13 (see results in Table 1 and Table 2).

Fig. 4 depicts the features selected when the proportion achieved the best performance on the RS126 dataset (i.e. using 50% of the features). It is clear that not all the features selected are close to the central residue and certain long distance positions (e.g. $d_r = 7$ and $d_r = 15$ in this experiment) are also important for the classification. Meanwhile, when analysing the residue patterns

[5] Best performance is achieved using about 50% of the features on the RS126 dataset; while this proportion increases to 57% on the CB513 dataset.

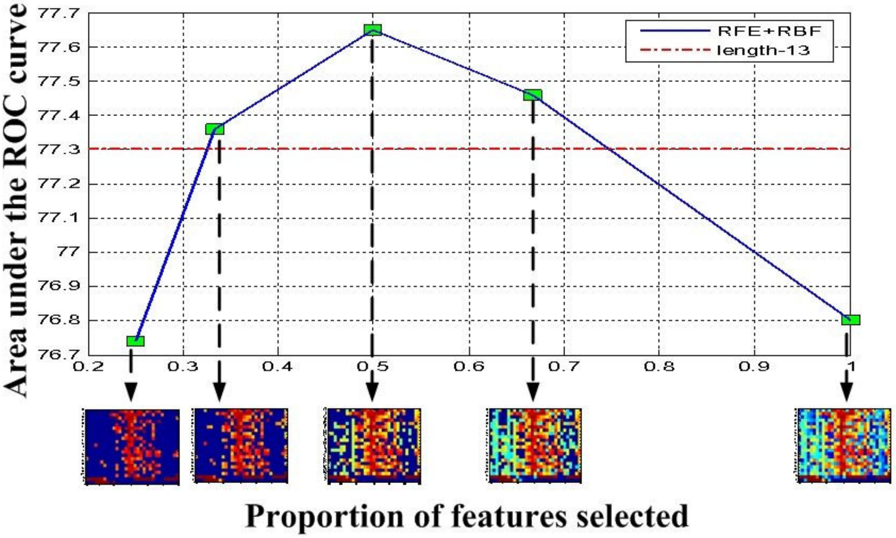


Fig. 3. Feature selection performance at various proportions of features used (RS126 dataset). From a window size of 31, best performance is achieved using about 50% of the features. While the shorter window considered (length-13) is also about 50%, feature selection selects those amino acid positions, consistent with the distribution observed in Fig. 1.

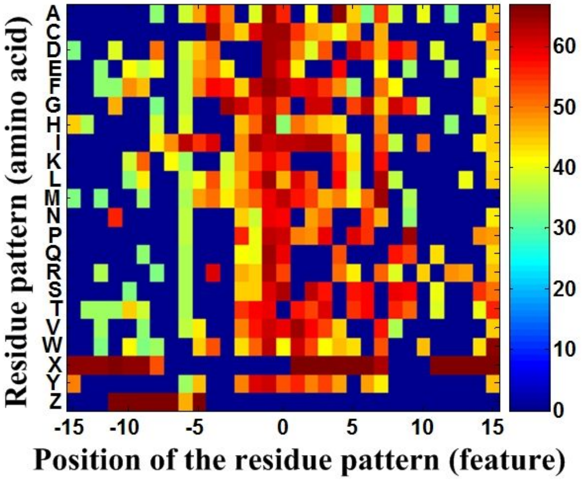


Fig. 4. Selection of relevant residue patterns (RS126 dataset). The relevance of each amino acid at each position with respect to the centre is shown as an intensity plot. Automatically selected features include amino acids known to have a bias towards β-sheet formation: D (Asp), F (Phe), G (Gly), I (Ile), L (Leu), M (Met), T (Thr), W (Trp) and X.

(amino acids), some patterns are shown to receive very popular votes. They are: D (Asp), F (Phe), G (Gly), I (Ile), L (Leu), M (Met), T (Thr), W (Trp) and X (unknown amino acids). This observation is consistent with some discussion in [4]:

- The frequency of observation of a hydrophobic amino acid (e.g. Ile, Leu, Met, Trp, Phe) one position before and one position after β-sheets is low. Therefore, when they appear very close to the central residue, the central residue is more likely to be $\sim$E.
- Asp and Gly tend to act as a β-sheet terminator and are therefore very important in formatting β-sheets. In similar fashion, Thr has high propensity for initiating a β-sheet and is also important for β-sheet formation.

In addition, another residue pattern: X (unknown amino acids) is also highly weighted in this experiment, although it was not analysed in [4]. The reason is that X is a rare feature, which appears only 11 times in the RS126 dataset. Moreover, all examples containing this pattern are in class $\sim$E, which explains why it is selected as an important residue pattern by RFE.

4 Conclusion and Future Work

Our observations show that some long range amino acid interactions can be captured in a feature reduction setting for improved prediction of β-sheet secondary structures. In the feature selection process, the top ranked amino acids are those that are specifically associated with the initiation and termination of β-sheet formations.

In the immediate future, we will verify that the prediction advantage we found for β-sheets is not observed when trying to classify α-helices from coil structures. We also intend formulating the prediction problem as a structured learning problem to exploit long-range dependencies in a principled manner, as for example in the phrase disambiguation task of machine translation [10].

References

1. Bamber, D.: The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. Journal of Mathematical Psychology 12, 387–415 (1975)
2. Cole, C., Barber, J., Barton, G.: The jpred 3 secondary structure prediction server. Nucleic Acids Research, doi:10.1093/nar/gkn238
3. Cuff, J., Barton, G.: Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. Proteins: Struct. Funct. Genet. 34, 508–519 (1999)
4. FarzadFard, F., Gharaei, N., Pezesnk, H., Marashi, S.: β-sheet capping: Signals that initiate and terminate β-sheet formation. Journal of Structure Biology 161(1), 101–110 (2008)
5. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46(1-3), 389–422 (2002)

6. Hua, S., Sun, Z.: A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach. J. Mol. Biol. 308(2), 397–407 (2001)
7. Kabsch, W., Sander, C.: Dictionary of protein secondary structure: pattern recognition of hydrogen bonded and geometrical features. Biopolymers 22, 2577–2637 (1983)
8. Minh, H.Q., Niyogi, P., Yao, Y.: Mercer's theorem, feature map, and smoothing. In: COLT, pp. 154–168 (2006)
9. Nguyen, M., Rajapakse, J.: Multi-class support vector machines for protein secondary structure prediction. Genome Informatics 14 (2003)
10. Ni, Y., Saunders, C., Szedmak, S., Niranjan, M.: The application of structure learning in natural language processing. Machine Translation (in Press)
11. Qian, N., Sejnowski, T.: Predicting the secondary structure of globular proteins using neural network models. J. Mol. Biol. 202, 865–884 (1988)
12. Qian, N., Sejnowski, T.: Predicting the secondary structure of globular proteins using neural network models. Journal of Molecular Biology 202(4), 865–884 (1988)
13. Rost, B.: Protein secondary structure prediction continues to rise. Journal of Structural Biology 134, 204–218 (2001)
14. Rost, B., Sander, C.: Prediction of protein secondary structure at better than 70% accuracy. J. Mol. Biol. 232, 584–599 (1993)
15. Ward, J., McGuffin, L., Buxton, B., Jones, D.: Secondary structure prediction with support vector machines. Bioinformatics 19(13), 1650–1655 (2003)

Alpha Helix Prediction Based on Evolutionary Computation

Alfonso E. Márquez Chamorro, Federico Divina,
Jesús S. Aguilar Ruiz, and Gualberto Asencio Cortés

School of Engineering, Pablo de Olavide University of Sevilla, Spain
{amarcha,fdivina,aguilar,guaasecor}@upo.es

Abstract. Multiple approaches have been developed in order to predict the protein secondary structure. In this paper, we propose an approach to such a problem based on evolutionary computation. The proposed approach considers various amino acids properties in order to predict the secondary structure of a protein. In particular, we will consider the hydrophobicity, the polarity and the charge of amino acids. In this study, we focus on predicting a particular kind of secondary structure: α-helices. The results of our proposal will be a set of rules that will identify the beginning or the end of such a structure.

Keywords: Protein Secondary Structure Prediction, α-helix, Evolutionary Computation.

1 Introduction

Bioinformatics has been described as the science of managing, mining, and interpreting information from biological sequences and structures [1]. Two important fields are considered in Bioinformatics: Genomics and Proteomics. Genomics is the study and analysis of the genomes of organisms, while Proteomics is defined as the characterization and identification of the proteins encoded in a genome.

Proteins are one of the basic components in all organisms. Proteins form the basis of cellular life since they significantly affect the structural and functional characteristics of different cells and genes. The structure of a protein is divided into four hierarchy levels. At the first level, proteins are composed of linear sequences of amino acids linked by natural peptide links. This is known as the primary structure of the protein.

The change in one amino acid in a critical area of the protein may alter the biological function, as the higher level structures of the proteins are determined by the primary structure. The secondary structure of a protein is the consequence of the polypeptide chain folding. At this level, some protein structures like α-helices, β-sheets, turns and coils are present. The tertiary structure is the three-dimensional shape of the chain, while the quaternary structure is the final three-dimensional structure composed by all polypeptides chains that form a protein [1,2].

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 358–367, 2010.

With the success of the genome sequence projects, the amount of available proteins sequences has increased dramatically. However, the number of protein structures available is relatively small. This is due to the difficulty of predicting the structures that a protein will assume based only on its amino acid sequence. This implies that it is crucial to develop computational methods for automatically predict the 3D structure of proteins from their sequences. Knowledge of protein structure has great importance to the development of new drugs.

The problem of protein secondary structure prediction (PSSP) consists in predicting the location of α-helices, β-sheets and turns from a sequence of amino acids without any knowledge of the tertiary structure of the protein. PSSP has received much attention lately, since knowledge of the location of the elements in secondary structure could be used by approximation algorithms to obtain the tertiary structure of the protein. Being able to predict, from the amino acid sequence, how a protein will fold, is one of the main open problems in computational biology.

Several methods were applied to the PSSP problem. These methods can be divided into two categories: statistical and soft computing approaches. Statistical methods are based on the calculation of amino acid probabilities to belong to a secondary structure motif [3,4,5]. Soft computing provides processing capabilities in order to solve the problem of PSSP. The most popular soft computing paradigms for PSSP are: artificial neural networks (ANNs) [6,7,8], evolutionary computation [9], nearest neighbors [10,11] and support vector machines (SVMs)[12,13]. Some soft computing methods used in this problem are focused on determining contact maps (distances) between amino acids residues of a protein sequence. When a contact map is defined, proteins can be fold and the tertiary structure can be obtained.

In this paper, we propose a method, based on an evolutionary algorithm (EA), to predict α-helices from sequences of amino acids. We believe that EAs are good candidate form tackling this problem. In fact, PSSP can be seen as a search problem, where the search space is represented by all the possible folding rules. Such a space is very complex, and has huge size. EAs have proven to be particularly good in this kind of domains, due to their search ability and their capability of escaping from local optima.

In our proposal, prediction is made *ab initio*, i.e., without any known protein structure as a starting template for the search. The prediction model will consist in rules that predict both the beginning and the end of the regions corresponding to an α-helix. Existing methods fail in the α-helix boundaries prediction [14]. In a future development of the algorithm, we also intend to evolve rules for predicting β-sheets.

Previously, some evolutionary approaches have been applied to secondary structure prediction. In [15], a torsion angle representation representation was used. Torsion angles, denoted as (Φ, Ψ), represent the atom position of an amino acid chain, determining the polypeptid arquitechture chain. A possible representantion can be $[(\Phi_1, \Psi_1)...(\Phi_n, \Psi_n)]$ where n represents the total number of residues in a protein. The values that Φ and Ψ can assume are limited, since atom colissions must be avoided according to Ramachandran chart [16]. In lattice models developed in [9], each element location can be represented as a vector $(x_1, y_1)...(x_n, y_n)$ where x and y are the coordinates of each amino acid in a 2-dimensional lattice (or three coordinates in a 3-dimensional lattice).

The rest of paper is organized as follow. In section 2, we discuss our proposal to predict protein secondary structure motifs. Section 3 provides the experimentation and the obtained results. Finally, in the last section, we draw some conclusions and analyze possible future works.

2 Our Proposal

In this section, we present our proposal for the prediction of α-helices. An α-helix corresponds to a subsequence of amino acids, as shown in figure 1. Each amino acid in the sequence is identified by its position, being amino acids in positions N-cap and C-cap those that immediately precede or follow the beginning or the end of the structure, respectively.

$$\boxed{N_{CAP}}\underbrace{\begin{array}{|c|c|c|c|c|c|c|}\hline N_1 & N_2 & N_3 & ... & C_3 & C_2 & C_1 \\ \hline\end{array}}_{\alpha-helix}\boxed{C_{CAP}}$$

Fig. 1. Relevant positions in an α-helix

Figure 2 represents our experimental procedure to predict protein secondary structure. First, the α-helix sequences are obtained from the Protein Data Bank (PDB) [17], as described in the following sections. These data constitute the training set. Then, our EA is applied and a set of rules are generated. We generate rules for predicting the beginning and the end of an α-helix separately. At the end of the EA, a set of rules will be extracted.

In the following we discuss the various solutions we adopted for what regards the fitness, the representation and the genetic operators used.

2.1 Encoding

In our approach, each individual may represent either the beginning or the end of an α-helix. Namely, each individual represents three properties of amino acids in positions N-cap, N1 or C1, C-cap. These are the limits of an α-helix sequence. The represented properties are hydrophobicity, polarity and charge. These properties have been shown to have certain relevance in PSSP [1,2]. We use Kyte-dolitle hydropathy profile for the hydrophobicity [18]. We have selected Grantham's profile [19] for polarity and Klein's scale for net charge [20]. The values of the properties are then normalized to a range of between -1 and 1 for hydrophobicity and polarity. Three values are used to represent the net charge of a residue: -1 (negative charge), 0 (neutral charge) and 1 (positive charge).

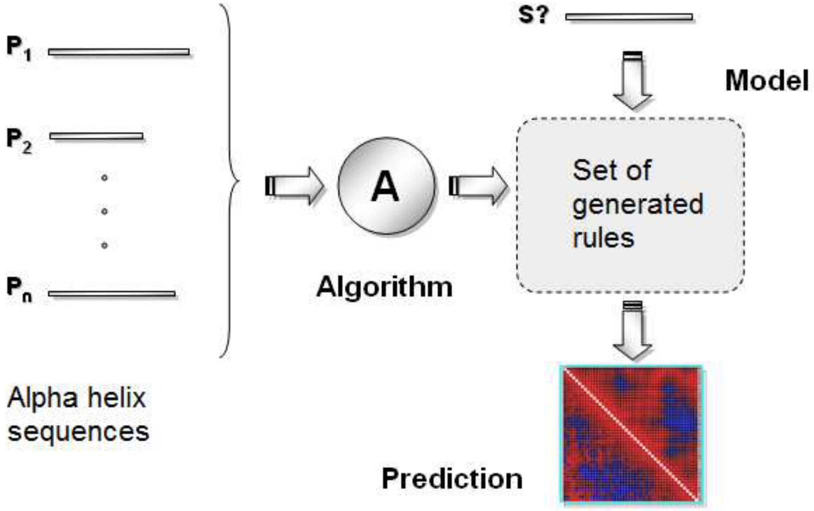


Fig. 2. Experimental and prediction procedure

$$\underbrace{\begin{array}{|c|c|c|c|c|}\hline P_1 & P_2 & P_3 & P_4 & P_5 \\ \hline\end{array}}_{N-Cap}\underbrace{\begin{array}{|c|c|c|c|c|}\hline P'_1 & P'2 & P'_3 & P'_4 & P'_5 \\ \hline\end{array}}_{N1}$$

Fig. 3. Example of chromosome codification for a beginning of an α-helix

So, for instance, in figure 3, positions P_1, P_2, P'_1, P'_2 represent the hidrophobicity values of the first and second amino acid respectively. Positions P_3, P_4, P'_3, P'_4 represent the polarity values according to Grant scale of the first and second amino acid respectively. Finally, positions P_5 and P'_5 represents the net charge property values of the two amino acids.

2.2 Fitness Function

The aim of the algorithm is to find both general and precise rules for identifying helices. To this aim, we have chosen as fitness of individuals the F-measure, which is given by the following formula:

$$F = 2 \cdot \frac{Recall \cdot Precision}{Recall + Precision}.$$

The higher the fitness, the better the individual. Recall represents the proportion of training examples that matches this rule. Precision represents the error rate.

Moreover, we also consider some physical-chemical properties (polarity and charge) information of the amino acids in positions N-Cap, N1 or C1, C-Cap, if the rule is relative to a beginning or an end of a helix, respectively. It has been demonstrated that there are molecules with asymmetrical distributions of charge in the limits of an α-helix [21]. This means that the residues in limits of the helix are polar, so the fitness of these individuals is increased. Moreover, in [22,23], it has been proven that many helices present a positive charge in its last turn and a negative charge at its first turn.

We increase the score of those individuals that fulfill one requirements in a 50%, and in a 100% for those individuals that present the two properties.

2.3 Genetic Operators

Individuals are selected with a roulette wheel mechanism. A roulette wheel is built, where the sector associated with each individual of the population is proportional its fitness. Individuals with higher fitnesses have more probability of being selected, having wider sectors associated to them.

Uniform crossover is used in order to generate offsprings. Crossover is applied with a 1.0 probability. All the offsprings are made by crossover except the one with best score which was copied without any change (elitism). Mutation is applied with a probability of 0.5. If mutation is applied, one gene of the individual is randomly selected, and its value is increased or decreased by 0.01. If the selected gene is relative to the charge of the amino acid, then its value is randomly changed to one of the other two allowed possibilities. After that an individual has been mutated, it is checked for validity, i.e., its values are within the ranges allowed for each properties. If the encoded rule is not valid, then the mutation is discarded.

The initial population is randomly initialized. After having evaluated the initial population, the first generation is created. If the fitness of the best individual does not increase for twenty generations, the algorithm is stopped and a solution is provided.

We evolve two populations separately: one population contains individuals that encode rules identifying the beginning of an α-helix, while the other population contains individuals representing rules for the end of the helix. At the end of the evolutionary process, the best individuals from each population are extracted, and together they form the proposed solution.

3 Experiments and Discussion

In this section, we present the experimentation performed in order to assess the validity of our proposal.

In order to test the proposed algorithm, we have used data obtained from PDB. Protein secondary structure is obtained from amino acid sequences, as well as the distances between pairs of amino acids. All this information is included in the PDB site. The Worldwide PDB [24], is an international collaboration organized by the processing and distribution of the PDB file. The online PDB file [17] is the repository that coordinates and related information on nearly $65,000$ structures ($65,378$ structures in May 18, 2010), including proteins, nucleic acids and complex macromolecules that have been obtained through techniques of X-ray crystallography, NMR (nuclear magnetic resonance) and electron microscope.

We have obtained a set of 12,830 non-homologous different protein sequences with an homology lower than 30%, using the PDB Advanced Search [25]. We have only selected the structures which contains protein chains and not DNA or RNA chains using the Macromolecule type option. We reject the redundant sequences. The complete list of the 12,830 PDB protein identifiers can be downloaded in [26]. We parsed the required information from PDB files. At the Secondary Structure Section of PBD, different α-helix sequences of each protein can be obtained with the HELIX command. Once we have located the motifs in the protein sequence, we extract from this sequence, the amino acids from N-cap to C-cap positions of the helix (figure 1), which are relevant positions in a α-helix [21]. We have selected a subset of 5,000 α-helices sequences from a subset of proteins with length less than 150 residues from these 12,830 proteins. Each of these 5,000 sequences includes a begining and an end of helix. Thus, we have 5,000 windows of two amino acids in C-cap, C1 positions and 5,000 windows of two amino acids in N1, N-cap positions. These sequences represent our training data. The average size of the α-helix sequences is 9.86 residues.

A 10-fold cross-validation has been applied. The data set is divided into 10 subsets, and the holdout method is repeated 10 times. Each time, one of the 10 subsets is used as the test set and the other 9 subsets are put together to form a training set. Then the average result across all 10 trials is computed.

For each fold, we obtained the confusion matrix. Each column of the matrix represents the number of true or false predictions of a class, and each row represents the number of real instances. More specifically, the matrix contains information about the True Negatives (TN), False Positives (FP), False Negatives (FN), and True Positives (TP). TN is the number of correct predictions for a negative case (no ends or beginnings), FP is the number of incorrect predictions for a positive case (ends or beginnings of an helix), FN is the number of incorrect predictions for a negative case (no ends or beginnings) and TP is the number of correct predictions for a positive case (ends or beginnings of an helix).

For each fold, we compute the following results:

- Recall represents the percentage of correctly identified positive cases. In our case, Recall indicates what percentage of motifs has been correctly identified.

$$Recall = \frac{TP}{TP + FN}.$$

- Precision is a measure of false positive rate. Precision reflects the number of real predicted examples.

$$Precision = \frac{TP}{TP + FP}.$$

- Specificity, or True Negative Rate, measures the percentage of correctly identified negative cases. In this case, Specificity reflects what percentage of no motifs has been correctly identified.

$$Specificity = \frac{TN}{TN + FP}.$$

- Accuracy is also calculated.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}.$$

At each execution, a model is obtained. This model consists of two rules, one that identifies the beginning of an α-helix, and the other that identifies the end of such a structure. Since the number of rules needed to provide the best results is unknown, we have performed different experiments with different number of runs of the algorithm, namely from 10 to 40. So, for instance, after the experiments with 10 runs, a model with twenty rules is obtained, where half of the rules represent the beggining of an α-helix and the other half represent the end.

Table 1 and 2 show the obtained results relative to the N-cap and C-cap prediction, respectively. The first column specifies the number of execution of the algorithm, the second column gives the average recall obtained. The third and fourth columns provide the average specificity and precision, respectively. The last column is relative to the average accuracy obtained. For each measure, the standard deviation is also provided.

From tables 1 and 2, it can be noticed that the model provided by the algorithm is always very accurate, in fact, the average accuracy obtained is very high in all the cases, being the average 0.99. The precision of the model is also satisfactory, with an average of 0.70. This means that model obtained commits few classification errors. The average recall is about 0.60 for beginnings and 0.58 for ends of helix, which represents a good result es well, and it means that on average, 60% of the α-helices are recognized as such. We can also notice

Table 1. Average results and standard deviation obtained for different number of executions of the algorithm for N-cap prediction

Executions	$Recall_{\mu\pm\sigma}$	$Spec._{\mu\pm\sigma}$	$Prec._{\mu\pm\sigma}$	$Accuracy_{\mu\pm\sigma}$
10	$0.5525_{\pm 0.0437}$	$0.9895_{\pm 0.0005}$	$0.6553_{\pm 0.0232}$	$0.9935_{\pm 0.0008}$
20	$0.6212_{\pm 0.1156}$	$0.9924_{\pm 0.0007}$	$0.6857_{\pm 0.0220}$	$0.9948_{\pm 0.0015}$
30	$0.6275_{\pm 0.0922}$	$0.9948_{\pm 0.0005}$	$0.7368_{\pm 0.0315}$	$0.9940_{\pm 0.0016}$
40	$0.6025_{\pm 0.0848}$	$0.9937_{\pm 0.0006}$	$0.7320_{\pm 0.0372}$	$0.9937_{\pm 0.0013}$

Table 2. Average results and standard deviation obtained for different number of executions of the algorithm for C-cap prediction

Executions	$Recall_{\mu\pm\sigma}$	$Spec._{\mu\pm\sigma}$	$Prec._{\mu\pm\sigma}$	$Accuracy_{\mu\pm\sigma}$
10	$0.5933_{\pm 0.0565}$	$0.9889_{\pm 0.0005}$	$0.6338_{\pm 0.0218}$	$0.9955_{\pm 0.0007}$
20	$0.5728_{\pm 0.1185}$	$0.9943_{\pm 0.0006}$	$0.6589_{\pm 0.0250}$	$0.9952_{\pm 0.0018}$
30	$0.5936_{\pm 0.0933}$	$0.9935_{\pm 0.0006}$	$0.6859_{\pm 0.0302}$	$0.9972_{\pm 0.0020}$
40	$0.5870_{\pm 0.0848}$	$0.9925_{\pm 0.0006}$	$0.7005_{\pm 0.0450}$	$0.9966_{\pm 0.0015}$

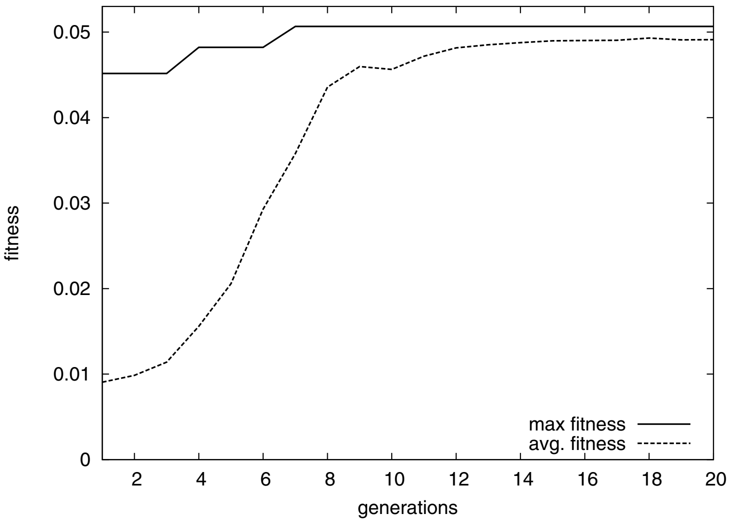


Fig. 4. Maximum Fitness vs. Average Fitness

that producing a model with more rules (the more executions the more rules will be part of the model produced) does not neccessarily help in increasing the precision. For the rest of the measures, the results become more or less stable after 20 executions of the algorithm.

Our algorithm is capable of producing satisfactory results using an elevated number of sequences (5,000 beginnings and 5,000 ends of helix sequences). This is, in our opinion, an important result, since the number of protein sequences available increase by the day, and thus, having a method that is scalable would be very important.

Other approaches were developed to predict starts of helix. The start position are correctly predicted for approximately 30% of all predicted helices in [14]. The number of correctly predicted alpha-helix start positions was improved from 30% to 38% in [27]. These results are widely exceeded by our approach, as our algorithm predicts about 60% of the start positions correctly. We have not found references for the C-cap helix prediction in literature.

It is also interesting to inspect the behavior of our EA. Figure 4 shows a graphical representation of the maximum and average fitness values at different generations relative to a typical run. We can notice that the maximum fitness is achieved very early, at about generation seven, and then it is stable. This may suggest that we should try to increase the mutation probability, or apply a mutation operator that introduces more changes in an individual, in order to increase diversity in the population. Another estrategy, could be to apply some local search method with a given probability. Such local search would help in improving the fitness of the individuals.

On the other hand, the average fitness increases constantly, and tends to converge to the maximum fitness toward the end of the run.

4 Conclusions and Future Work

In this paper, we have proposed an evolutionary algorithm for the prediction of α-helix motifs in protein sequences. The algorithm incorporates in the fitness three amino acids properties: hydrophocity, polarity and net charge. These properties have been shown to be relevant in the determination of the beginning and end of helices, and thus they helped to improve the search process performed by the algorithm.

We have performed experiments using a set of 5,000 α-helix sequences extracted from a protein data set from Protein Data Bank composed by 12,830 non-redundant and non-homologous protein with an homology rate lower than 30%. To the best of our knowledge, no other approaches have used such an high number of sequences in α-helix capping regions prediction. Results obtained on this data set are encouraging and in particular, the accuracy characterizing the prediction models obtained is very high independently from the number of rules generated.

As for future development, we are analyzing different properties to be included in the fitness function in order to increase the quality of the prediction model. Moreover, we are studying the possibility of incorporating a local search phase that will help to improve individuals. We also intend to extend our experimentation to other datasets of protein sequences and we want to expand the number of residues in the window of amino acids. Finally, we also want to produce a model for the prediction of both α-helices and β-sheets.

Acknowledgements

This research was supported by the Project of Excellence P07-TIC-02611 “Sistemas Inteligentes para descubrir patrones de comportamiento. Aplicación a base de datos biológicas” and by Spanish Ministry of Science and Technology under grants TIN2007-68084-C02-00 and by the Junta de Andalucia, Project P07-TIC-02611.

References

1. Gu, J., Bourne, P.E.: Structural Bioinformatics (Methods of Biochemical Analysis). Wiley-Blackwell, Chichester (2003)
2. Berg, J.M., Stryer, L.: Biochemistry. Reverte (2008)
3. Chou, P.Y., Fasman, G.D.: Prediction of protein conformation. Biochemistry 13(2), 222–245 (1974)
4. Garnier, J., Osguthorpe, D.J., Robson, B.: Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. J. Mol. Biol. 120, 97–120 (1978)
5. Lim, V.I.: Algorithms for prediction of a-helical and b-structural regions in globular proteins. J. Mol. Biol. 88, 857–872 (1974)
6. Qian, N., Sejnowski, T.J.: Predicting the secondary structure of globular proteins using neural network models. J. Mol. Biol. 202, 865–884 (1988)

7. McGuffin, L.J., Bryson, K., Jones, D.T.: The psipred protein structure prediction server. Bioinformatics 16, 404–405 (2000)
8. Fariselli, P., Casadio, R.: A neural network based predictor of residue contacts in proteins. Protein Engineering 12, 15–21 (1999)
9. Unger, R., Moult, J.: Genetic algorithms for protein folding simulations. Biochim. Biophys. 231, 75–81 (1993)
10. Frishman, D., Argos, P.: Incorporation of non-local interactions in protein secondary structure prediction from the amino acid sequence. Protein Engineering 9, 133–142 (1996)
11. Salamov, A.A., Solovyev, V.V.: Protein secondary structure prediction using local alignments. J. Mol. Biol. 268, 31–36 (1997)
12. Ward, J.J., McGuffin, L.J., Buxton, B.F., Jone, D.T.: Secondary structure prediction with support vector machines. Bioinformatics 13, 1650–1655 (2003)
13. Cheng, J., Baldi, P.: Improved residue contact prediction using support vector machines and a large feature set. Bioinformatics 8, 113 (2007)
14. Wilson, C.L., Hubbard, S.J., Doig: A critical assessment of the secondary structure prediction of alpha-helices and their n-termini in proteins. Protein Eng. 15, 545–554 (2002)
15. Cui, Y., Chen, R.S., Hung, W.: Protein folding simulation with genetic algorithm and supersecondary structure constraints. Proteins: Structure, Function and Genetics 31, 247–257 (1998)
16. Ramakrishnan, C., Ramachandran, G.N.: Stereochemical criteria for polypeptide and protein chain conformation. Byophys Journal 5, 909–933 (1965)
17. Protein data bank online repository, ftp://ftp.wwpdb.org
18. Kyte, J., Doolittle, R.F.: A simple method for displaying the hydropathic character of a protein. J. J. Mol. Bio. 157, 105–132 (1982)
19. Grantham, R.: Amino acid difference formula to help explain protein evolution. J. J. Mol. Bio. 185, 862–864 (1974)
20. Klein, P., Kanehisa, M., DeLisi, C.: Prediction of protein function from sequence properties: Discriminant analysis of a data base. Biochim. Biophys. 787, 221–226 (1984)
21. Richardson, J.S., Richardson, D.C.: Amino acid preferences for specific locations at the ends of alpha helices. Science 240, 1648–1652 (1998)
22. Doig, A.J.: Baldwin R.L. N- and c-capping preferences for all 20 amino acids in alpha-helical peptides. Protein Science 4(7), 1325–1336 (1995)
23. Fonseca, N.A., Camacho, R., Magalhaes, A.L.: Amino acid pairing at the n- and c-termini of helical segments in proteins. Proteins 70, 188–196 (2007)
24. Protein data bank web, http://www.wwpdb.org
25. Protein data bank advanced search, http://www.pdb.org/pdb/search/advSearch.do
26. Complete list of pdb protein identifiers used in this article, http://www.upo.es/eps/marquez/proteins.txt
27. Wilson, C.L., Boardman, P.E., Doig, A.J., Hubbard, S.J.: Improved prediction for n-termini of alpha-helices using empirical information. Proteins 57(2), 322–330 (2004)

An On/Off Lattice Approach to Protein Structure Prediction from Contact Maps

Stefano Teso, Cristina Di Risio, Andrea Passerini, and Roberto Battiti

Dipartimento di Ingegneria e Scienza dell'Informazione
Università degli Studi di Trento, Italy
{teso,dirisio,passerini,battiti}@disi.unitn.it

Abstract. An important unsolved problem in structural bioinformatics is that of protein structure prediction (PSP), the reconstruction of a biologically plausible three-dimensional structure for a given protein given only its amino acid sequence. The PSP problem is of enormous interest, because the function of proteins is a direct consequence of their three-dimensional structure. Approaches to solve the PSP use protein models that range from very realistic (all-atom) to very simple (on a lattice). Finer representations usually generate better candidate structures, but are computationally more costly than the simpler on-lattice ones. In this work we propose a combined approach that makes use of a simple and fast lattice protein structure prediction algorithm, REMC-HPPFP, to compute a number of coarse candidate structures. These are later refined by 3Distill, an off-lattice, residue-level protein structure predictor. We prove that the lattice algorithm is able to bootstrap 3Distill, which consequently converges much faster, allowing for shorter execution times without noticeably degrading the quality of the predictions. This novel method allows us to generate a large set of decoys of quality comparable to those computed by the off-lattice method alone, but using a fraction of the computations. As a result, our method could be used to build large databases of predicted decoys for analysis, or for selecting the best candidate structures through reranking techniques. Furthermore our method is generic, in that it can be applied to other algorithms than 3Distill.

Keywords: Protein Structure Prediction, HP model, Contact Maps, Simulated Annealing, Replica Exchange Monte Carlo.

1 Introduction

Protein structure prediction (PSP) is the problem of inferring the tertiary structure of proteins given only information on their primary structure. This problem is of the highest importance for several reasons: the function of a protein is strictly tied to its three-dimensional structure, but the experimental determination of the tertiary structure is still a complex, time consuming and expensive process. In addition, in some cases it is impossible to obtain structural information with experimental techniques: many proteins are too large for NMR analysis and some classes of proteins such as membrane ones are very difficult

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 368–379, 2010.

to crystallize for X-ray diffraction [4]. As a matter of fact, most of the known protein sequences are not yet assigned a corresponding structure: in spite of the long-standing community-wide effort, most known proteins still lack a resolved structure, and of the nearly two million protein sequences currently known, fewer than 2% have an associated structure.

Following the idea that similar sequences are bound to represent similar structures [6], at least at a local level, comparative modeling methods have been developed which exploit local homology information to compute the structure of novel query protein sequences. Remote homology techniques are employed for fold recognition when sequence conservation is lacking. However, when no close or remote homologues are available, these methods cannot be applied, and *de novo* structure prediction must be performed.

The current methods for *de novo* PSP can be split in roughly three groups. A first group accounts for all-atom molecular simulation methods, which try to mimic the physical folding process starting from first principles. They have huge computational requirements and have not been very successful for realistically sized proteins. A second group includes all those methods that search the space of atom- or residue-level conformations for a native-like fold using some more or less empirical energy function to assess the quality of the candidate structures. Meta-heuristic optimization algorithms are usually employed to perform the search. These methods have had much more success than the ones in the previous class, but they are still very computationally expensive. Finally, a third group includes methods that rely on residue-level (or coarser) structure representations and enforce them to lie on a regular lattice, embedded in purely synthetic force fields. Methods in this group can find a native-like decoy with relatively less computational effort than methods in the former groups, but the resulting structures are not as realistic. They are typically used as tools to analyze the statistical properties of the folding landscape, rather than to generate reliable structures.

In this work we present a novel method that combines the complementary strengths of off-lattice empirical models and on-lattice ones, and allows to generate a large number of comparatively good quality decoys with a fraction of the computational power required by standard methods. The underlying idea is to combine two existing *de novo* PSP algorithms: a modified version of REMC-HPPFP [15], a fast prediction method based on a very coarse structure representation, which is used to compute a first set of rough decoys; 3Distill [1], a more realistic method that uses a finer structure representation, which is employed to refine the decoys generated by REMC-HPPFP. Albeit based on a simple idea, we prove that our method is indeed able to combine the features of REMC-HPPFP and 3Distill: it generates competitive structures with much less effort. Furthermore, the idea underlying our method is generic, meaning that it is in no way restricted to 3Distill, and may prove useful to improve the efficiency of other fine-grained structure prediction algorithms.

This paper is structured as follows. In Section 2 we review some of the relevant methods for the protein structure problem. In Section 3 we describe our combined

PSP approach and the two methods on which it is based. In Section 4 we describe the experiments carried out to benchmark our method and compare it to the baseline. In Section 5 we discuss the results of the experiments and show that our method is ultimately successful in reducing the amount of computation required. Finally, in Section 6 we draw the conclusions on this work and describe some future research directions.

2 Related Work

De novo PSP methods include both off-lattice and on-lattice models and methods. In off-lattice models, the residues are free to be placed at arbitrary continuous coordinates in the three-dimensional Euclidean space. The simplest way to represent residues is as hard spheres of fixed radius centered on the C_α atom, but other more complex representations are available as well. In other, intermediate models, all the atoms of the backbone are modeled, but the side chain is represented as a hard sphere centered at the center of mass of the real side chain. It has been noted however that the lower computational demands of coarse-grained models does not necessarily come at the cost of inferior expressiveness [10].

In on-lattice models the protein conformation is restricted, such that each residue occupies a different vertex on a lattice. Consecutive residues in the primary structure are placed at adjacent positions, and the protein chain becomes itself a self-avoiding path on the lattice. Lattice models employ a variety of two- and three-dimensional lattice: square, triangular, cubic, face-centered cubic, diamond, and others with very high degrees of freedom. For the representational power of different common and less-common lattices, we refer the reader to [11]. On-lattice models have been chiefly used as tools for studying protein folding, because the simplified representation allows for an easier mathematical treatment [10].

Common approaches to the PSP problem include the aggregation of short structural fragments, for instance Rosetta [12], and the use of contact maps [16,1]. We focus on the latter approach. The idea is to split the prediction task into two simpler sub-tasks: first generate *de novo* an accurate, residue-by-residue contact map from the protein sequence, and then reconstruct the protein structure from the contact map. This is a sound approach, as contact maps can be shown to encode the same information as the structure they represent [16]. To date, a few contact map predictors have been proposed: SVMcon [2], Xxstout [1], and NNcon [14] among others. As for the reconstruction process itself, a popular approach is to use some form of stochastic optimization, as in the seminal paper by Vendruscolo et al. [16] and 3Distill [1].

3 Method

Our proposed method is based on two well known existing *de novo* PSP algorithms: in the next couple of sections we will introduce them and explain their pros and weaknesses.

3.1 3Distill

An often advocated approach to the PSP is to split the main *de novo* structure prediction problem into a set of simpler prediction tasks. Distill [1] is a hierarchy of state-of-the-art prediction servers that follows this approach. The Distill servers compute a number of one-dimensional features (such as secondary structure, solvent accessibility, and contact density) and two-dimensional features (such as fine and coarse contact maps, coarse protein topology). The main idea is that all servers make use of features predicted in the lower levels of the hierarchy, starting from the primary structure, to predict more complex features.

At the top of the hierarchy, the 3Distill server computes the protein tertiary structure, as a residue-level C_α trace, given predicted features from all the other servers. A preliminary implementation of 3Distill took part to the CASP 6 competition [8] and was ranked among the best 20 predictors out of 181 on Novel Fold hard targets and Near Novel Fold targets. 3Distill was chosen because it is simple and relatively fast when compared to other *de novo* algorithms.

The main feature input into 3Distill is a predicted (multi-class) contact map, which specifies a set of soft physical constraints for all pairwise inter-residue distances. For a detailed description of contact maps, see [16]. Other input features include a predicted per-residue secondary structure and a predicted coarse-grained contact maps, which defines the appropriate distances between pairs of secondary structure elements. To avoid the computational burden of all-atom models, 3Distill relies on a reduced backbone-only protein model. Furthermore, residues that are predicted to belong to an α-helix are modeled as rigid, ideal helices. This solves the problem of folding the helices during the optimization stage, and decreases the complexity of the conformational search. To mimic the minimal observed distance between atoms of different amino acids, the volume of each C_α is modeled as a hard sphere of radius 5.0 $\mathring{A}$, and the distance between consecutive residues is set to 3.8 $\mathring{A}$. These values were rigorously inferred from statistical analysis of real world data [1].

All candidate conformations have an associated pseudo-potential that is defined in terms of the input contact maps and secondary structure. The energy of a conformation estimates how much it violates the constraints imposed by the given fine and coarse contact maps, while at the same time penalizing non-physical configurations (i.e., overlapping or too far away residues). For an in detph description, see [1].

The mechanism used by 3Distill to search for the native conformation is Simulated Annealing (SA) [7]. SA is an iterative procedure: starting from a random candidate structure, at each iteration it perturbates the structure producing another candidate configuration. The newly generated configuration replaces the old one if it is better (has a lower energy), with probability one; or if it is worse (higher energy) with a probability that depends on the magnitude of the energy difference. This second condition is controlled by a so called temperature parameter: when the temperature is high, even very bad configurations have a high probability of having accepted; when it is low, almost all worsening configurations are rejected. In 3Distill the temperature decreases linearly with the number of iterations, meaning that as the search proceeds the temperature moves towards zero and the probability of accepting worsening moves goes to zero as well. For further details, we refer the reader to [7].

In 3Distill, each iteration of SA traverses the whole structure, perturbating each residue in the order in which it appears in the protein chain. A perturbation amounts to displacing a residue according to the following rules: (1) If the residue is neither an endpoint nor in a helix, it is rotated by a random angle around the segment joining its two neighboring residues. (2) If the residue is an endpoint of the chain and not part of a helix, it is rotated at random around its only neighbor. (3) If the residue is part of a helix, the whole helix is rotated at random. This set of moves guarantees 3Distill to efficiently explore the conformational space. We note that each traversal of the protein structure amounts to h perturbations, where h is the overall number of free residues (not in a helix) and helices. The SA algorithm stops after a given amount of traversals.

3.2 REMC-HPPFP

The Hydrophobic-Polar model (HP model for short) [3] is a very basic model of protein folding based on a reduced, residue-level representation of the tertiary structure. In this model, proteins are represented as backbone-only configurations and the residues are forced to lie on a regular, typically cubical lattice, with no overlap. In the HP model, each residue is either hydrophobic (H) or polar (P). The HP model is designed to capture the fact that folding is mainly driven by hydrophobic interactions between the residues. Following this idea, the energy of a configuration $\mathbf{x}$ is defined empirically in terms of neighboring residues: two residues are called *topological neighbors* if they are not consecutive in the protein sequence and share an edge of the lattice. The energy associated to an HP configuration is the negated number of topological neighbors that are both hydrophobic. In other words, this energy function favors those configurations containing a densely packed core of hydrophobic residues. Solving an HP problem instance involves finding the native conformation, that is, the structure having the lowest possible associated energy.

Despite its simplicity, the HP model has been proven to be NP-complete in both two and three dimensions on the cubic lattice [7, 14], and NP-hard on a general lattice [21], including the face-centered cubic and triangular lattices. For this reason, HP model solvers usually resort to heuristic optimization algorithms to search the conformational space. REMC-HPPFP [15] is one of the state-of-the-art solvers of square and cubic lattice HP instances. It makes use of a very effective stochastic search procedure, named Replica Exchange Monte Carlo (REMC for short) that is especially geared towards high-dimensional optimization problems. REMC-HPPFP has been shown to lead to superior results with respect to competing methods, such as PERM [5] and ACO-HPPFP-3 [13] in a set of synthetic and on biologically-derived benchmark instances [15]. The core features are the REMC optimization heuristic and the set of moves used to perform the search itself. We briefly discuss them in the following, see [15] for details.

The REMC search heuristic is reminiscent of Simulated Annealing, in that a candidate protein structure is perturbated at each iteration, by applying a random move, and the resulting structure is accepted or rejected depending on the energy delta with respect to the old configuration. However in this case, multiple configurations, called replicas, are optimized concurrently. Each configuration has its own fixed temperature, which does *not* decrease with time. Replicas are indexed from 1 to m, and the temperature of each replica is a monotonically increasing function of its index. Once every k iterations, with k a fixed parameter, the energy of adjacent replicas is compared, and if certain energy conditions are met, the two replicas are exchanged, meaning that the ith replica will become the $(i+1)$th and vice versa. This way the replicas change temperature based on their energy level. The set of moves used by REMC-HPPFP to perturbate the candidate configurations comprises a set of standard residue by residue moves, termed VHSD moves, and the non-standard pull move [9]. This set of moves is the most complete and efficient set of moves available to date for the HP model on the square and cubic lattices.

3.3 On/Off Lattice Cascade

The main issue with 3Distill is that, even being one of the simplest *de novo* predictors proposed, the conformational space is huge and requires a large amount of computational power to find low energy configurations. This is a common problem for all fine-grained structure predictors. On the other hand, the REMC-HPPFP algorithm shows very good performances on HP instances. Our primary aim in this work is to combine the efficiency of on-lattice methods with the accuracy of off-lattice models. We do so by first using a suitably modified version of REMC-HPPFP to quickly produce a candidate on-lattice structure that (partially) satisfies a given residue-level contact map, and then refining the obtained structure by using 3Distill with the same contact map. The intermediate lattice structures generated by the modified REMC-HPPFP can be thought as bootstrapping 3Distill, by making it start its search from more favorable regions of the search space.

To obtain the best results from the cooperation of REMC-HPPFP and 3Distill, we had to implement a new lattice energy function. The new function defines the fitness of a configuration in terms of how much it satisfies a given multi-class contact map. The formal definition is as follows:

$$E(\mathbf{x}; C, p, k) = \sum_{i,j} E(d_{ij}; c_{ij}, p, k)$$

$$E(d_{ij}; c_{ij}, p, k) = \begin{cases} |d_{ij} - \tau_c|^p & \text{if } d_{ij} < \tau_c \\ |d_{ij} - \tau_{c+1}|^p & \text{if } d_{ij} > \tau_{c+1} \\ -k & \text{otherwise} \end{cases}$$

where $\mathbf{x}$ is a candidate protein structure, $C = [c_{ij}]$ is a multi-class contact map, with each class c having range $[\tau_c, \tau_{c+1}]$, and d_{ij} is the Euclidean distance between residues i and j. The pairwise energy potential is a polynomial of the difference between the actual distance between residues i and j and the closest threshold of the predicted contact class. The two constants p and k are parameters used to adjust the energy function to the data at hand. In particular, k defines the net gain for a satisfied contact, and p controls the amount of penalty for an unsatisfied contact. In this new model, structures lie on a cubic three-dimensional lattice of fixed side 3.8 $\mathring{A}$, the same as the default inter-residue distance for 3Distill.

To summarize, our method consists of a modified REMC-HPPFP version that, by virtue of a new energy function, is able to find on-lattice configurations that best satisfy a given residue-level contact map. Aside from the new energy function, the REMC-HPPFP algorithm is unchanged. This novel method is used to generate one or more lattice configurations, which are then refined with 3Distill; both algorithms use the same predicted contact map. All in all, the new cascade method requires four additional parameters to be specified: p and k shape the energy function, the other two are T_1 and T_2, the number of iterations to run the on-lattice and off-lattice algorithms for, respectively.

To allow for a common measurement unit of computation, we define the concept of *big iteration* as a complete traversal of the protein structure by the search algorithm. For 3Distill a big iteration involves h structure perturbations, each requiring to compute the value of the energy function for the newly generated configuration, for a total of $h = O(n)$ energy computations. For REMC-HPPFP, a big iteration equates to $n \times m$ structure perturbations, where m is the number of replicas, again amounting to $O(mn) = O(n)$ energy updates. The computational complexity of the two algorithms is thus $O(n^2)$ per big iteration, as both require $O(n)$ pseudo-instructions for each energy function evaluation.

4 Experiments

The goal of the experiments is to assess the ability of our combined method to generate decoys of quality comparable to that of the original 3Distill algorithm, and to evaluate the amount of computation required to attain such decoys. The quality of the decoys is defined in terms of the TM-score [17] to the experimentally determined native fold. TM-score values range in $[0, 1]$, with all values larger than 0.4 suggesting a topologically correct prediction, and for all scores above 0.7 a good structural superposition between the predicted and the native folds.

The tests are based on a dataset of 171 proteins with no detected homology, with length between 50 and 200 residues. The contact maps were predicted by Xxstout [1] with threshold values $\tau_1 = 8$Å, $\tau_2 = 13$Å, and $\tau_3 = 19$Å using a recursive neural network while exploiting evolutionary information in the form of multiple alignment profiles, plus the contact map of the nearest template when available. All template-matching qualities and all relevant SCOP classes (all α, all β, α/β, $\alpha+\beta$, coiled-coil, and small) are represented in this data set. The data was kindly provided by the Distill team.

4.1 Selection of the Lattice Energy Function

The goal of the first batch of experiments was to tune the parameters p and k of the new lattice energy function to maximize the TM-score of the resulting decoys as expected. During previous experiments we observed that the quality of 3Distill results is positively correlated to the TM-score of the inputs structures, and the same holds for its convergence. The dataset is varied enough to guarantee that the parameters p^* and k^* found are generalizable to other data sets. During these experiments, for simplicity we kept the other parameters of the modified REMC-HPPFP fixed to values used in the original paper for the three-dimensional lattice [15]. In particular, the number of replicas is two.

For this set of experiments, we sampled the performance of the modified REMC-HPPFP for (p, k) values taken from a grid in the (p, k) parameter space. A preliminary set of runs was performed on a small subset of protein instances to determine the extents of the grid, for a total of 10 structures for each proteins, 100 iterations each. We found some reasonable values to be $p \in [0.25, 2.25]$ (at increments of 0.50) and $k = \{0, 10, 100, 1000\}$. Outside this range, the performance of our method degraded quickly. The grid itself is uniform in the p dimension and exponential in k: the reason is that p appears as an exponent in the energy function, while k is an additive linear term. Next we performed a thorough exhaustive search: for each (p, k) value in the grid, now with p increments of 0.25, we ran our modified REMC-HPPFP on all proteins in the dataset, 100 runs per protein, 100 iterations per run, and compared the average TM-score of the generated decoys. Using this method, the best parameters were found to be $p^* = 1.75$ and $k^* = 0$.

4.2 Behavior over Time

Given the optimal values p^* and k^*, we evaluated the number of big iterations (T_1, T_2) that our method needs to obtain results comparable to those of 3Distill alone. To compare the performance of our combined approach to 3Distill, we use the ratio between the TM-score reached by our algorithm and the best TM-score obtained by 3Distill alone. We defined a uniform grid in the (T_1, T_2) parameter space. The upper bound for $T_2 < 5000$ was determined experimentally by observing the number of big iterations needed to achieve pseudo-convergence with 3Distill. For T_1 we just used the same number of iterations defined in the original paper, $T_1 < 100$.

In all the runs, the lattice algorithm was run with the same parameters as in the previous set of experiments, together with the newly found p and k. The parameters of 3Distill were setup as in [1]. We ran the combined algorithm for all proteins in the dataset, 100 runs for each protein, with $T_1 < 100$ and $T_2 < 5000$, recording the intermediate candidate structures during the optimization procedures, so to properly fill in the (T_1, T_2) grid.

For every protein and (T_1, T_2) pair, we computed the average TM-score of the predicted folds and normalized it with respect to the average TM-score of the structures for the same protein found at $(T_1, T_2) = (0, 5000)$. We call this quantity the “quality ratio”, i.e., the ratio between the TM-score for proteins found by our method using (T_1, T_2) iterations, and the TM-score of the structures predicted by 3Distill. Then for each point in the (T_1, T_2) grid we computed the average of the structure quality ratio over all decoys and over all proteins.

We note that the number of energy evaluations per big iteration in 3Distill is equal to the number of control points h, whereas for REMC-HPPFP it is equal to the number of residues n for each replica. Despite being both asymptotically $O(n)$, in practice these two quantities are not identical. This makes it difficult to experimentally compare the values of T_1 and T_2, because h is a structural property depending on the predicted protein secondary structure. Hence h may be different between proteins of the same size. To account for this fact, we split the results by protein length in 3 different classes, with ranges from 50 to 200: the first class contains proteins of length from 50 to 99, the second those of length from 100 to 149, the third those of length from 150 to 200. For each class we computed the average number of hinges $\hat{h}$ and the average number of residues $\hat{n}$, and rescaled the T_2 axis by $\hat{n}/\hat{h}$. This results in 3 grids, shown in Figure 1.

5 Discussion

The main result of this paper is that, in all the plots, the combined algorithm is shown to be able to produce structures of quality comparable to that of 3Distill alone, but with a far smaller number of energy evaluations. Multiple combinations of T_1, T_2 show this behavior. Generally, it can be observed that: (a) To obtain structures of quality ratio at least 0.7, that is, structures whose quality is comparable to that of structures found by a full run ($T_2 = 5000$) of the costly off-lattice algorithm, it is sufficient to use $T_1 = 100$ and $T_2 \leq 500$. This amounts to one about tenth of the energy evaluations. (b) To obtain structures of quality ratio at least 0.9, that is, structures whose quality is indistinguishable from that of structures found with $(T_1 = 0, T_2 = 5000)$, roughly 100 on-lattice iterations followed by 2000 off-lattice iterations are sufficient. This amounts to less than one half of the energy evaluations. Thus employing an on-lattice search strategy to obtain initial candidate configurations actually improves the search speed of the off-lattice algorithm.

One surprising result, implicit in the previous discussion, is that the on-lattice algorithm can generate structures of good quality, with respect to those found by the off-lattice method. This can be seen by observing the curves at all grid points with $T_2 = 0$. It follows that despite its simplicity, the cubic lattice, when paired with our contact map driven energy function, is able to model topologically correct, even if coarse, decoys. This seem to support the idea that the on-lattice algorithm is able to bootstrap 3Distill in a region of the search space that contains native-like folds.

Finally, the plots show that the quality ratios reported at the curves with T_1=fixed improve monotonically with respect to T_1. This means that allowing for increasing amounts of on-lattice search, and consequently for better initial candidates to the off-lattice algorithm, helps the latter. This proves that it is

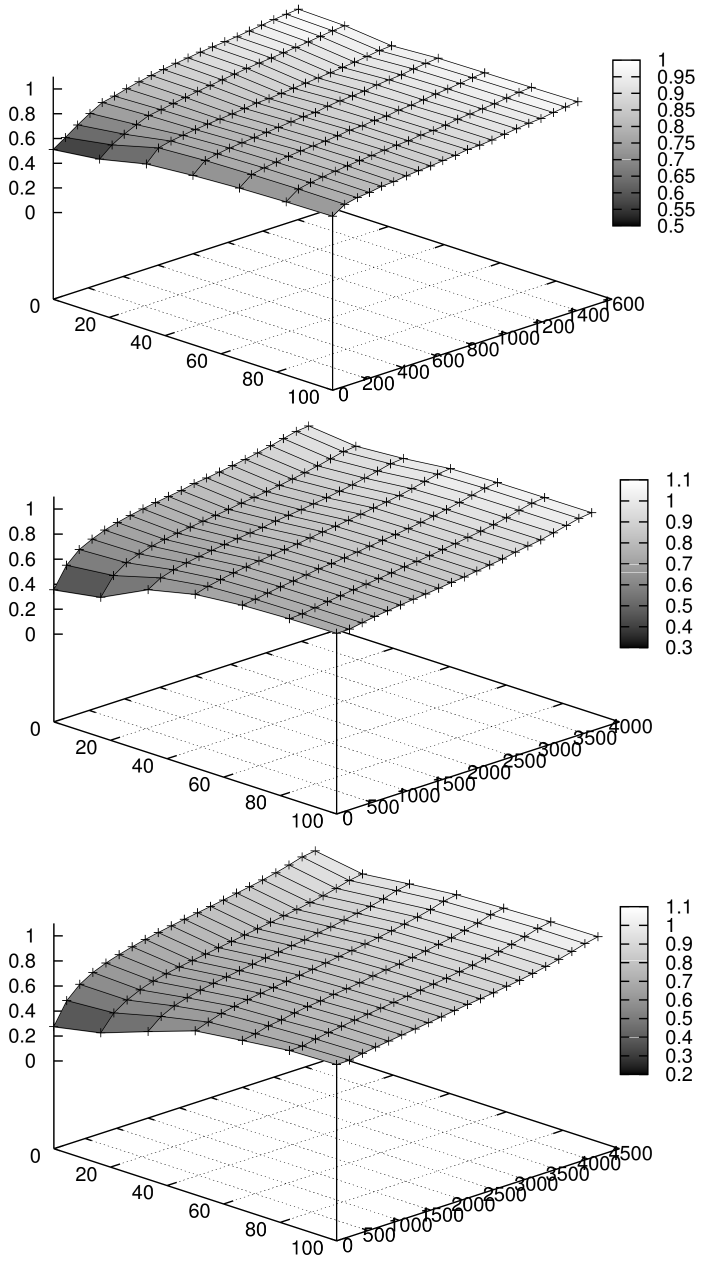


Fig. 1. Each plot represents the behavior over time of our combined method. The axes represent T_1 and T_2 and the height of each point represents the average solution quality ratio (over all decoys and all proteins in the dataset) described in Section 4.2. The upper plot refers to proteins of length 50 to 99 residues; the middle plot to proteins of length 100-149; the last one to proteins of length 150-200.

the on-lattice to be ultimately responsible for enhancing the convergence speed of 3Distill, and not some random external factor such as a different distribution of the initial configurations. The curves with T_2=fixed instead appear to reach convergence at $T_1 = 100$. This validates our choice of $T_1 \leq 100$, and shows that increasing its value would not improve the performance of the lattice algorithm any further.

We note, however, that running the combined algorithm with both T_1 and T_2 set to the maximum values does not significantly improve upon the solutions found by the off-lattice algorithm alone. A possible explanation is that, simply, 3Distill has already reached convergence and that it would be unable to do better than it actually is even when initialized with a good candidate structure.

Summarizing, the above results show that our novel combined on/off lattice approach to protein structure prediction indeed requires potentially fewer energy evaluations to generate good quality, low energy decoys for proteins of length less than 200 residues. This enables for reduced execution time and an increased throughput of structure prediction whenever a contact map is given. The key point is that the resulting pool of structures will probably contain some native-like folds. Ultimately, the higher throughput of our method can serve two purposes: firstly, producing a large population of decoys for statistical analysis; and secondly, to apply reranking techniques with an improved likelihood of finding native-like structures. The ranking approach is very interesting, because it is possible to tune our algorithm with small $(T1, T2)$ values and be able to select very good decoys with small computational effort.

6 Conclusions

In this work we presented a method that combines two existing state-of-the-art approaches to the Protein Structure Prediction problem in a novel way by exploiting the complementary strengths of the two. In particular, a lattice algorithm is used to quickly construct a number of coarse, yet relatively good quality, decoys from predicted contact maps; an off-lattice algorithm is later employed to refine the search. Thanks to the lower number of degrees of freedom, the on-lattice search effectively acts as a bootstrapping step for 3Distill, which converges much faster since the starting candidate conformation is already located in a favorable region of the search space. We proved experimentally that the proposed method allows to generate structures of quality comparable to those generated by 3Distill alone with a fraction of the computational effort. The improvement amounts to one order of magnitude less evaluations of the energy potential, which is the most computationally intensive part of most search algorithms. We stress that our approach is not restricted to 3Distill at all, and that other fine-grained *de novo* algorithms could benefit from it as well. The proposed method potentially allows to build large databases of decoys for analysis or for the later application of reranking techniques to determine the most plausible native folds.

Acknowledgments

The authors would like to thank Gianluca Pollastri for sharing the 3Distill source code.

References

1. Baú, D., Martin, A.J.M., Mooney, C., Vullo, A., Walsh, I., Pollastri, G.: Distill: a suite of web servers for the prediction of one-, two- and three-dimensional structural features of proteins. BMC bioinformatics 7(1), 402 (2006)
2. Cheng, J., Baldi, P.: Improved residue contact prediction using support vector machines and a large feature set. BMC bioinformatics 8(1), 113 (2007)
3. Dill, K.A.: Theory for the folding and stability of globular proteins. Biochemistry 24(6), 1501–1509 (1985)
4. Garavito, R.M., Picot, D., Loll, P.J.: Strategies for crystallizing membrane proteins. Journal of bioenergetics and biomembranes 28(1), 13–27 (1996)
5. Hsu, H.P., Mehra, V., Nadler, W., Grassberger, P.: Growth-based optimization algorithm for lattice heteropolymers. Physical Review E 68(2), 21113 (2003)
6. Kaczanowski, S., Zielenkiewicz, P.: Why similar protein sequences encode similar three-dimensional structures? Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)
7. Kirkpatrick, S.: Optimization by simulated annealing: Quantitative studies. Journal of Statistical Physics 34(5), 975–986 (1984)
8. Kryshtafovych, A., Venclovas, C., Fidelis, K., Moult, J.: Progress over the first decade of CASP experiments. Proteins: Structure, Function, and Bioinformatics 61(S7), 225–236 (2005)
9. Lesh, N., Mitzenmacher, M., Whitesides, S.: A complete and effective move set for simplified protein folding. In: Proceedings of the seventh annual international conference on Research in computational molecular biology, p. 195. ACM, New York (2003)
10. Oakley, M.T., Barthel, D., Bykov, Y., Garibaldi, J.M., Burke, E.K., Krasnogor, N., Hirst, J.D.: Search strategies in structural bioinformatics. Current Protein and Peptide Science 9(3), 260–274 (2008)
11. Pierri, C.L., De Grassi, A., Turi, A.: Lattices for ab initio protein structure prediction. Proteins: Structure, Function, and Bioinformatics 73(2), 351–361 (2008)
12. Rohl, C.A., Strauss, C.E.M., Misura, K., Baker, D.: Protein structure prediction using Rosetta. Methods in enzymology, 66–93 (2004)
13. Shmygelska, A., Hoos, H.H.: An ant colony optimisation algorithm for the 2 D and 3 D hydrophobic polar protein folding problem. BMC bioinformatics 6(1), 30 (2005)
14. Tegge, A.N., Wang, Z., Eickholt, J., Cheng, J.: NNcon: Improved Protein Contact Map Prediction Using 2D-Recursive Neural Networks. Nucleic Acids Research (May 2009)
15. Thachuk, C., Shmygelska, A., Hoos, H.H.: A replica exchange Monte Carlo algorithm for protein folding in the HP model. BMC bioinformatics 8(1), 342 (2007)
16. Vendruscolo, M., Kussell, E., Domany, E.: Recovery of protein structure from contact maps. Folding and Design 2(5), 295–306 (1997)
17. Zhang, Y., Skolnick, J.: Scoring function for automated assessment of protein structure template quality. PROTEINS-NEW YORK- 68(4), 1020 (2007)

Part VI

Protein Protein Interaction and Network Inference

Biological Protein-Protein Interaction Prediction Using Binding Free Energies and Linear Dimensionality Reduction

L. Rueda[1], Carolina Garate[2], Sridip Banerjee[1], and Md. Mominul Aziz[1]

[1] School of Computer Science, University of Windsor, 401 Sunset Ave.,
Windsor, ON, N9B 3P4, Canada
{lrueda,banerje1,azizc}@cs.uwindsor.ca

[2] Department of Computer Science, University of Concepcion, Concepcion, Chile
cgarate@udec.cl

Abstract. An important issue in understanding and classifying protein-protein interactions (PPI) is to characterize their interfaces in order to discriminate between transient and obligate complexes. We propose a classification approach to discriminate between these two types of complexes. Our approach uses contact and binding free energies of the residues present in the interaction, which are the input features for the classifiers. A total of 282 features are extracted for each complex, and the classification is performed via recently proposed dimensionality reduction (LDR) methods, including the well-know Fisher's discriminant analysis and two heteroscedastic approaches. The results on a standard benchmark of transient and obligate protein complexes show that LDR approaches achieve a very high classification accuracy (over 78%), outperforming various support vector machines and nearest-neighbor classifiers. An additional insight on the proposed approach and experiments on different subsets of features shows that solvation energies can be used in the classification, leading to a performance comparable to using the full binding free energies of the interaction.

Keywords: protein-protein interaction, classification, binding free energy, linear dimensionality reduction.

1 Introduction

Protein-protein interaction (PPI) is involved in multiple cellular processes such as signal transduction, immune response, regulation of gene expression, and different processes where the oligomerization is a requirement to achieve a biologically active state. In this context, interactions can be attractive or repulsive, which may result in the formation of intermolecular clusters or aggregates. Although PPI depends on the protein surfaces and on the environmental conditions, many efforts have been made to understand the factors responsible for interactions between proteins at the atomic level [1,2,3]. PPI has been studied from many different perspectives and for different purposes. According to [4], prediction of protein interactions can be focused on three main goals: (i) predicting

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 383–394, 2010.

the interfaces involved in the interaction, (ii) predicting the spatial arrangement of the interacting chains or molecules, and (iii) predicting the identity of the molecules involved in the interaction. One typical case of the latter main goal is to differentiate between specific types of PPI, namely obligate versus transient interactions, i.e., interactions that can be identified by its duration. Characterizing PPI in terms of specific goals including prediction of different types can be carried out in many different ways and using many different descriptors or features [5], including solvent accessibility, residual vicinity, shape of the structure of the interface, secondary structure, planarity, conservation scores, physicochemical features, hydrophobicity electrostatic and solvation energies, just to mention a few. In this work, we focus on using energetic features.

Some of the studies in PPI consider the characterization of the geometry [6], physicochemical properties [7], the preference of residues to appear on the surface [8], and the role of hydrogen bridges, saline bridges and hydrophobic and polar interactions on the proteins surfaces [9]. Other studies include the analysis of the loss of surface accessible to solvent [10] as a result of the interaction and the analysis of the conservation of residues in the interaction surface [11]. In an upper level, amino acids composition of protein-protein interfaces have been studied to infer the composition of the residues at the interface, which is generally different from the rest of the surface. A comprehensive study was conducted by the authors of [12], who studied six types of interfaces: intra and inter domains, homo and hetero-oligomers, and obligate and transient complexes. That study concluded that the amino acid composition of these surfaces are different, as there is only 1.5% of similarity between the internal and external surfaces, and 0.2% similarity between hetero surfaces belonging to obligate homo complex and transient homo complexes. They found, on the other hand, a 16.3% similarity between homo and hetero complexes.

To study the behavior of transient and obligate interactions, in [13], a classification of these two types of interactions was proposed, where interactions are classified based on the lifetime of the complex. Obligate interactions are usually more stable, while transient interactions are less stable and, hence, more difficult to discriminate and understand, due to their short life [14]. Protomers from obligate complexes do not exist as stable structures in vivo, whereas protomers of non-obligate complexes may dissociate from each other and stay as stable and functional units. For these reasons, it is one of the prime importance of proteomics to distinguish between obligate and transient complexes. Additionally, in [15], it was proposed that interfaces in obligate complexes are inherently hydrophobic. Another work that deserves attention is that of Zhu et al. [16], in which three different types of interaction are studied, namely crystal packing, obligate and non-obligate interactions. Their study is based on using solvent accessible surface area, conservation scores, and shapes of the interfaces.

The interfaces of some transient complexes were also found to be with clusters of hydrophobic residues [17]. Moreover, they are rich in aromatic residues and arginine but depleted in other charged residues [18]. However, hydrophobicity at the interfaces of transient complexes is not as distinguishable from the remainder of the surface as hydrophobicity at the interfaces of the obligate complexes [18]. As a result, it is difficult to make an accurate prediction of the interfaces of transient complexes using a single parameter of residue interface propensity.

In [19], a research on protein-protein interactions was conducted in which each interaction is analyzed in physical interaction, co-complex relationship and co-member of the pathway (i.e. enzymes are involved in enzyme or metabolic ways). This study attempted to determine the accuracy of predictions of interactions, applying six different classification methods, namely random forest (RF), RF-based k-NN, Bayes, decision trees, logistic regression, and support vector machines (SVM). RF was shown to be the most robust and efficient method among the six aforementioned approaches for predicting protein-protein interactions. While this study concluded that the co-complex relationship is the easiest to predict, the situation could change when larger datasets are available.

Although interfaces have been the main subject of study to predict protein-protein interactions, an accuracy of 70% has been independently achieved by several different groups [20,21,22,23]. These approaches have been carried out by analyzing a wide range of parameters, including solvation energies, amino acid composition, conservation, electrostatic energies, and hydrophobicity. In a recent work, prediction of four different PPI types has been performed, including transient Enzyme inhibitor/Non Enzyme inhibitor and permanent homo/hetero obligate complexes [24]. That work uses association rules to understand and characterize the diverse kinds of interactions, and carry out experiments on 147 pre-classified complexes (a smaller set than the one used in [25], and which is used here).

In this paper, a classification approach to predict transient and obligate protein-protein interactions is proposed. We use heteroscedastic linear discriminant analysis as the primary classification method, which is discussed in Section 2. For each protein complex, its three-dimensional structure, obtained from the Protein Data Bank (PDB) [26], is processed to extract binding free energies, namely solvation and electrostatic, producing as many as 282 features. The details of this process are discussed in Section 4. Other two classifiers, namely the k-nearest neighbor and a support vector machine, are also used for experimental comparison (briefly discussed in Section 3). Experiments on more than 400 transient and obligate complexes on two different datasets show a high accuracy in classification, more than 78% – the discussions of these experiments are in Section 5. Further analysis on the results demonstrate that solvation energies are crucial in distinguishing transient and obligate complexes, and using these features solo leads to a performance comparable to, if not better than, using the full binding free energies of the interaction.

2 Linear Dimensionality Reduction

In this section, we discuss the homoscedastic and heteroscedastic classifiers used in our approach. Linear dimensionality reduction (LDR) is a well-studied topic in the field of pattern recognition. The basic idea of LDR is to represent an object of dimension n as a lower-dimensional vector of dimension d, achieving this by performing a lineal transformation. The advantage of using a linear transformation is that, although the derivation of the underlying transformation may be slower, the classification is extremely fast as it performs linear-time operations to reduce to dimensions, typically, much lower than the original one.

We consider two classes, ω_1 y ω_2, represented by two normally distributed random vectors $\mathbf{x}_1 \sim N(\mathbf{m}_1, \mathbf{S}_1)$ and $\mathbf{x}_2 \sim N(\mathbf{m}_2, \mathbf{S}_2)$, respectively, with p_1 and p_2 the *a priori* probabilities. After the LDR is applied, two new random vectors $\mathbf{y}_1 = \mathbf{A}\mathbf{x}_1$ and $\mathbf{y}_2 = \mathbf{A}\mathbf{x}_2$, where $\mathbf{y}_1 \sim N(\mathbf{A}\mathbf{m}_1; \mathbf{A}\mathbf{S}_1\mathbf{A}^t)$ and $\mathbf{y}_2 \sim N(\mathbf{A}\mathbf{m}_2; \mathbf{A}\mathbf{S}_2\mathbf{A}^t)$ with $\mathbf{m}_i$ and $\mathbf{S}_i$ being the mean vectors and covariance matrices in the original space, respectively. The aim is to find a linear transformation matrix $\mathbf{A}$ in such a way that the new classes ($\mathbf{y}_i = \mathbf{A}\mathbf{x}_i$) are as separable as possible. Various criteria have been proposed to measure this separability [27]. We consider three LDR methods: (a) the well-know Fisher's discriminant analysis (FDA) [28,29], a recently proposed heteroscedastic discriminant analysis (HDA) approach [30], and the even more recent Chernoff discriminant analysis (CDA) approach [27] – a brief discussion of these three follows.

Let $\mathbf{S}_W = p_1\mathbf{S}_1 + p_2\mathbf{S}_2$ and $\mathbf{S}_E = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t$ be the within-class and between-class scatter matrices respectively. The well-known FDA criterion consists of maximizing the Mahalanobis distance between the transformed distributions by finding $\mathbf{A}$ that maximizes the following function [28]:

$$J_{FDA}(\mathbf{A}) = tr\left\{(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1}(\mathbf{A}\mathbf{S}_E\mathbf{A}^t)\right\} . \tag{1}$$

The matrix $\mathbf{A}$ that maximizes (1) is obtained by finding the eigenvalue decomposition of the matrix:

$$\mathbf{S}_{FDA} = \mathbf{S}_W^{-1}\mathbf{S}_E , \tag{2}$$

and taking the d eigenvectors whose eigenvalues are the largest ones. Since $\mathbf{S}_E$ is of rank one, $\mathbf{S}_W^{-1}\mathbf{S}_E$ is also of rank one. Thus, the eigenvalue decomposition of $\mathbf{S}_W^{-1}\mathbf{S}_E$ leads to only one non-zero eigenvalue, and hence FDA can only reduce to dimension $d = 1$.

HDA has been recently proposed as a new LDR technique for normally distributed classes [30], which takes the Chernoff distance in the original space into consideration to minimize the error rate in the transformed space. It can be seen as a generalization of FDA to consider heteroscedastic classes, and the aim is to obtain the matrix $\mathbf{A}$ that maximizes the function:

$$\begin{aligned} J_{HDA}(\mathbf{A}) = tr\Big\{&(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1}\big[\mathbf{A}\mathbf{S}_E\mathbf{A}^t \\ &-\mathbf{A}\mathbf{S}_W^{\frac{1}{2}}\tfrac{p_1\log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_1\mathbf{S}_W^{-\frac{1}{2}})+p_2\log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_2\mathbf{S}_W^{-\frac{1}{2}})}{p_1p_2}\mathbf{S}_W^{\frac{1}{2}}\mathbf{A}^t\big]\Big\} \end{aligned} \tag{3}$$

where the logarithm of a matrix $\mathbf{M}$, $\log(\mathbf{M})$, is defined as:

$$\log(\mathbf{M}) \triangleq \boldsymbol{\Phi}\log(\boldsymbol{\Lambda})\boldsymbol{\Phi}^{-1} . \tag{4}$$

with $\boldsymbol{\Phi}$ and $\boldsymbol{\Lambda}$ representing the eigenvectors and eigenvalues of $\mathbf{M}$, respectively.

The solution to this criterion is given by computing the eigenvalue decomposition of:

$$\begin{aligned}&\mathbf{S}_{HDA} = \mathbf{S}_W^{-1}\\ &\left[\mathbf{S}_E - \mathbf{S}_W^{\frac{1}{2}} \frac{p_1 \log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_1\mathbf{S}_W^{-\frac{1}{2}}) + p_2 \log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_2\mathbf{S}_W^{-\frac{1}{2}})}{p_1 p_2} \mathbf{S}_W^{\frac{1}{2}}\right]\end{aligned} \tag{5}$$

and choosing the d eigenvectors whose corresponding eigenvalues are the largest ones.

CDA is an LDR method that has been recently proposed, and its aim is to maximize the separability of the distributions in the transformed space measured by the Chernoff distance between the two classes. CDA assumes that the classes are normally distributed (in the original and transformed spaces), maximizing the following function [27]:

$$\begin{aligned}J_{CDA}(\mathbf{A}) = tr\{&p_1 p_2 \mathbf{A}\mathbf{S}_E\mathbf{A}^t(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1}\\ &+ \log(\mathbf{A}\mathbf{S}_W\mathbf{A}^t) - p_1\log(\mathbf{A}\mathbf{S}_1\mathbf{A}^t) - p_2\log(\mathbf{A}\mathbf{S}_2\mathbf{A}^t)\}\end{aligned} \tag{6}$$

where $\mathbf{S}_W = p_1\mathbf{S}_1 + p_2\mathbf{S}_2$, $\mathbf{S}_E = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t$.

It has been shown in [27] that for any normally distributed random vectors, $\mathbf{x_1}$ and $\mathbf{x_2}$, there always exists an orthogonal matrix $\mathbf{Q}$, where $\mathbf{Q}\mathbf{Q}^t = \mathbf{I}$, such that $J_{CDA}(\mathbf{A}) = J_{CDA}(\mathbf{Q})$ for any $\mathbf{A}$ or rank d. Thus, without loss of generality, here, we assume that $\mathbf{A}$ is an orthogonal matrix. In [27], a gradient-based algorithm was proposed, which maximizes the function (6) in an iterative way. The algorithm starts with an arbitrary orthogonal matrix $\mathbf{A}^{(1)}$, and at step $k+1$, $\mathbf{A}^{(k+1)}$ is computed as follows:

$$\mathbf{A}^{(k+1)} = \mathbf{A}^{(k)} + \alpha_k \nabla J_{CDA}(\mathbf{A}^{(k)}) \tag{7}$$

where the gradient for J_{CDA} is:

$$\begin{aligned}&\tfrac{\partial J_{CDA}}{\partial \mathbf{A}} = \nabla J_{CDA}(\mathbf{A}) = 2p_1p_2\left[\mathbf{S}_E\mathbf{A}^t(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1}\right.\\ &\left.-\mathbf{S}_W\mathbf{A}^t(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1}(\mathbf{A}\mathbf{S}_E\mathbf{A}^t)(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1}\right]^t\\ &+2\left[\mathbf{S}_W\mathbf{A}^t(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1} - p_1\mathbf{S}_1\mathbf{A}^t(\mathbf{A}\mathbf{S}_1\mathbf{A}^t)^{-1}\right.\\ &\left.-p_2\mathbf{S}_2\mathbf{A}^t(\mathbf{A}\mathbf{S}_2\mathbf{A}^t)^{-1}\right]^t\end{aligned}$$

For this gradient algorithm, a learning rate, α_k needs to be computed. In order to ensure that the gradient algorithm converges, α_k needs to be maximized. In [27], the secant method is proposed for this, and the aim is to maximize the function:

$$\phi_k(\alpha) = J_{CDA}(\mathbf{A}^{(k)} + \alpha\nabla J_{CDA}(\mathbf{A}^{(k)})) \tag{8}$$

Starting with two initial values $\alpha^{(0)}$ and $\alpha^{(1)}$, the value of $\alpha^{(j+1)}$ at time $j+1$ is iteratively found as follows:

$$\alpha^{(j+1)} = \alpha^{(j)} + \frac{\alpha^{(j)} - \alpha^{(j-1)}}{\frac{d\phi_k}{d\alpha}(\alpha^{(j)}) - \frac{d\phi_k}{d\alpha}(\alpha^{(j-1)})}\frac{d\phi_k}{d\alpha}(\alpha^{(j)}) \tag{9}$$

where

$$\frac{d\phi_k}{d\alpha}(\alpha) = [\nabla J_{CDA}(\mathbf{A}^{(k)} + \alpha \nabla J_{CDA}(\mathbf{A}^{(k)}))] \cdot \nabla J_{CDA}(\mathbf{A}^{(k)}) \tag{10}$$

The operator "$\cdot$" is the dot product between two matrices, and is computed, for any two matrices $\mathbf{C}$ and $\mathbf{D}$, as $\mathbf{C} \cdot \mathbf{D} = tr\{\mathbf{C}\ \mathbf{D}\}$. The value of $\nabla J_{CDA}(\mathbf{A}^{(k)} + \alpha \nabla J_{CDA}(\mathbf{A}^{(k)}))$ is computed by replacing $\mathbf{A}$ for $\mathbf{A}+\alpha\nabla J_{CDA}(\mathbf{A})$ in the equation (8).

Finally, with the definition of $\frac{d\phi_k}{d\alpha}(\alpha)$, Equation (9) can be solved, and the gradient algorithm continues with the next iteration. The complete algorithm can be found in [27]. One of the keys in this algorithm is the initialization of the matrix $\mathbf{A}$, and in this work, we have performed ten different initializations and then chosen the solution for $\mathbf{A}$ that gives the maximum Chernoff distance.

3 Other Classifiers

In order to compare the LDR methods with other benchmarks, a classification was performed with two other state-of-the-art classifiers: k-nearest-neighbor (k-NN) and an SVM. For the k-NN classifier, six different distance functions were implemented, namely angle, Chebychev, Euclidean, Manhattan, Minkowski and Pearson correlation. For each distance function, different values of $k = 1, ..., 20$ were evaluated, where the maximum value of $k = 20$ was taken roughly from $\sqrt{N}$ with N being the total number of complexes. The resulting accuracies were evaluated to observe the best overall performance of each distance, and hence we chose the Euclidean distance. The resulting accuracies of k-NN with the Euclidean distance, and the best value of k from 1 to 20 are reported in Section 5.

For the SVM classifier, different kernels were implemented and evaluated using the OSU-SVM toolbox in Matlab [31]. Three different types of kernels were implemented, namely polynomial, radial basis function (RBF), and sigmoid. For the polynomial kernel, polynomials of degree $p = 2, 3, ..., 8$ were considered. For the RBF, the parameters C and γ were optimized using grid search. As in k-NN, these different classifiers were evaluated and the maximum accuracy for all datasets resulted from the RBF kernel, with the parameters C and γ optimized. These results are reported on the fifth column of both Tables.

4 Protein-Protein Interaction Classification

To begin the classification process two dataset of transient and obligate complexes were obtained from previous works of [25] and [16]. Two types of complexes were classified as one of two classes: transient or obligate. Each complex is listed in the form of one or more chains for ligand and receptor respectively. The relevant data about the structure of the complex was obtained from the Protein Data Bank (PDB) [26]. When more than one chain are present on either ligand or receptor, they are merged into a single one, producing a complex with two interacting chains, one for the ligand and another for the receptor.

Obtaining binding free energies, even for a single complex, may take a considerable amount of time. Thus, for this purpose feature extraction is performed using FastContact [32], an approach that obtains a fast estimate of the binding free energy based on a statistically determined solvation contact potential and Coulomb electrostatics with a distance-dependent dielectric constant. The interaction between two chains is estimated as the sum of the standard intermolecular Coulombic electrostatic potential ($4r$ used as the distance-dependent dielectric constant), plus the most essential features of solvation free energy that includes hydrophobic interactions. For each complex, FastContact delivers the electrostatic energy, solvation free energy, and the top 20 maximum and minimum values (along with the corresponding residue number and amino acid) for: (i) residues contributing to the binding free energy, (ii) ligand residues contributing to the solvation free energy, (iii) ligand residues contributing to the electrostatic energy, (iv) receptor residues contributing to the solvation free energy, (v) receptor residues contributing to the electrostatic energy, (vi) receptor-ligand residue solvation constants, and (vii) receptor-ligand residue electrostatic constants.

For each complex, all energy values (minimum and maximum) were obtained as indicated in (i)-(vii). Thus, all these values (with the residue numbers) and the total solvation and electrostatic energy values compose a total of 282 features.

Due to the large number of features present in most datasets, compared to the number of samples, problems of dimensionality arise. More precisely, ill-conditioned matrices would be present when applying LDR methods, and hence principal component analysis is applied to each dataset by removing all components which are less than 10^{-5} times the largest eigenvalue of the within-class scatter of the dataset.

In order to classify each complex, first a linear algebraic operation $\mathbf{y} = \mathbf{Ax}$ is applied to the n-dimensional vector, obtaining $\mathbf{y}$, a d-dimensional vector, where d is ideally much smaller than n. The linear transformation matrix $\mathbf{A}$ corresponds to the one obtained by either of the LDR methods discussed in Section 2. The resulting vector $\mathbf{y}$ is then passed through a quadratic Bayesian (QB) classifier [28], which is the optimal classifier for normal distributions.

5 Experimental Results

To create the datasets for classification, two pre-classified datasets of protein complexes ware obtained from the studies of [25] and [16]. The first set of proteins, Mintseris et al. dataset, contains complexes of two classes: 209 transient complexes and 115 obligate complexes. The second dataset, Zhu et al. dataset, contains 62 transient complexes and 75 obligate complexes as two different classes for classification. The main datasets were created by retrieving each complex from PDB, and then obtaining the 282 features by invoking FastContact, as discussed in Section 4.

To study the effects of the different types of energies and ligand/receptor, we created a total of 13 different subsets of features for each dataset including:

Table 1. Results of classification accuracy for the 13 PPI subsets extracted from Mintseris et al. dataset [25], using different LDR methods and a comparison with k-NN and SVM

				QB					
Subset	n	k-NN	SVM	FDA	d^*	HDA	d^*	CDA	d^*
All Energetic	282	76.38	77.30	70.38	1	77.50	4	76.87	9
Binding Free Energy	40	69.94	72.09	71.86	1	75.20	6	73.33	5
Ligand Energy	80	72.70	74.54	66.58	1	76.42	8	76.44	6
Ligand Solvation	40	77.91	75.46	69.65	1	75.81	2	74.86	8
Ligand Electrostatic	40	69.33	70.86	72.09	1	72.14	7	71.84	4
Receptor Energies	80	74.54	74.23	67.17	1	76.42	6	76.74	11
Receptor Solvation	40	75.46	75.46	68.73	1	75.50	1	74.60	3
Receptor Electrostatic	40	72.09	70.55	68.47	1	69.71	3	70.31	12
Ligand-Receptor Energies	80	71.78	71.78	67.91	1	75.94	7	75.32	7
Ligand-Receptor Solv.	40	72.09	70.55	65.64	1	71.84	9	72.76	4
Ligand-Receptor Elect.	40	73.62	74.85	72.78	1	75.48	20	75.50	13
Solvation	120	78.53	76.07	65.72	1	76.70	14	76.41	11
Electrostatic	120	71.78	71.17	65.72	1	76.70	14	76.41	11

all 282 values, binding free energies, ligand/receptor solvation/electrostatic energies, ligand-receptor solvation and electrostatic energies, and solvation and electrostatic energies. The 13 datasets along with a short description in column one are listed in Tables 1 and 2. The second column lists the number of features of each dataset. As discussed earlier, PCA was applied to some datasets to avoid ill-conditioned matrices.

To study the performance of the classifiers, a 10-fold cross validation procedure was carried out, and then the average accuracy was computed, where accuracy for each individual fold was computed as follows: $acc = (TP + TN)/N_f$, where TN and TP are the true positive (obligate) and true negative (transient) counters, and N_f is the total number of complexes in the test set of the corresponding fold.

For the LDR schemes, three different classifiers were implemented and evaluated, namely the combinations of three LDR criteria discussed in Section 2, FDA, HDA and CDA, combined with a quadratic Bayesian (QB), implemented as discussed in Section 4. Note that we have also tested the classification with a linear Bayesian classifier, which yielded much lower classification accuracies than the QB. Then, only the results for QB are reported. For each of these classifiers reduction to dimensions $d = 1, ..., 20$ were performed, followed by QB. The dimensions that resulted in the best average accuracy for the 10-fold cross validation for each classifier are listed in the tables in the subsequent columns. Each column reports the highest average accuracy among all possible reduced dimensions, as well as the dimension in which the best accuracy is obtained, namely d^*. Since the classification problem is two-class, FDA always leads to

Table 2. Results of classification accuracy for the 13 PPI subsets extracted from Zhu et al. dataset [16], using different LDR methods and a comparison with k-NN and SVM

				QB					
Subset	n	k-NN	SVM	FDA	d^*	HDA	d^*	CDA	d^*
All Energetic	282	67.15	65.69	58.62	1	65.08	1	69.12	15
Binding Free Energy	40	64.96	59.85	55.59	1	65.74	9	63.23	7
Ligand Energy	80	68.61	69.34	60.05	1	70.60	15	72.08	5
Ligand Solvation	40	70.80	70.80	62.35	1	70.64	18	69.26	6
Ligand Electrostatic	40	64.23	62.77	49.55	1	60.51	4	59.58	3
Receptor Energies	80	65.69	67.88	52.54	1	68.86	13	72.79	19
Receptor Solvation	40	76.64	64.96	66.05	1	74.03	11	73.97	13
Receptor Electrostatic	40	61.31	64.96	54.95	1	65.48	6	67.48	5
Ligand-Receptor Energies	80	67.15	67.88	67.22	1	69.16	17	70.97	5
Ligand-Receptor Solv.	40	70.8	70.07	70.71	1	72.08	10	72.18	18
Ligand-Receptor Elect.	40	61.31	55.47	60.27	1	66.72	16	67.54	17
Solvation	120	73.72	71.53	51.41	1	65.33	6	75.41	7
Electrostatic	120	69.34	72.99	53.61	1	63.53	14	64.10	1

reducing to dimension one. The best accuracy for each method for each dataset is underlined to indicate the classifier that performed best of all for that dataset.

For the Mintseris et al. dataset (Table 1), it is clearly observable that the best performance was achieved by LDR methods combined with the QB classifier. Of these, the LDR criterion that achieves the best performance is HDA in as many as 6 out of 13 cases. Also, the classification of all LDR methods achieves the best performance in most of the cases, 10 out of 13 cases. This demonstrates that LDR methods perform better than k-NN and SVM. On the other hand, k-NN achieves better performance in more cases than the SVM, even though the results of these two are comparable in most of the cases. Regarding individual subsets, we observe that the best overall classification performance, 78.53%, was achieved by k-NN on Solvation energies. A comparison with other subsets, such as All Energetic, suggests that using a subset of features, such as Solvation energies, achieves an even better classification performance. In terms of energetic values, solvation leads to better performance than electrostatic values. This suggests that solvation is more important in classifying transient and obligate complexes. Additionally, using Solvation energies from the ligand only (just 40 features) leads to a classification accuracy of 77.91%, achieved by k-NN, which is no less than 1% below the best overall accuracy, obtained from all solvation values.

For the Zhu et al. dataset (Table 2), we observe that the best overall performance is delivered, again, using Solvation energies only, leading to an accuracy of 75.41%, which is achieved by CDA. Moreover, using the Solvation energies of the receptor only leads to an accuracy of 74.03%, slightly below that of using all Solvation energies. For this dataset, CDA is the best performer, yielding the highest accuracy in 8 out of 13 subsets. Again, as in Mintseris et al. dataset, the LDR methods perform much better than k-NN and SVM, and the Solvation energies by themselves can differentiate between the two types of complexes.

A final analysis of the results is done on the power of dimensionality reduction of the schemes. We observe that the best overall classification accuracy was obtained by HDA and CDA, while reducing from dimensions 120 to 14 and 11. In Zhu et al. dataset, the best classification accuracy achieved by CDA is 75.41%, while reducing from dimension 120 to 7. This thus implies not only a gain in classification accuracy but also in terms of classification speed. Similar results can be observed in the other cases, and hence demonstrating the power and simplicity of LDR schemes in this classification problem. To conclude, we emphasize that using a subset of features tends to be more productive than using all features, and hence demonstrating that the approach of considering different subsets of features leads to feature selection methods, even though more sophisticated approaches for feature selection could be used [33], a problem that is currently being investigated.

6 Conclusion

We have proposed a classification approach for transient and obligate protein-protein complexes. We have used linear dimensionality reduction (LDR) that involve homoscedastic and heteroscedastic criteria coupled with a quadratic Bayesian classifier. The results on two datasets of pre-classified complexes show that the LDR schemes coupled with QB achieves the best overall classification performance, even better than k-NN and an SVM with an RBF kernel. Comprehensive tests have been carried out in as many as 13 subsets of different features and for each dataset, showing that the best classification performance is achieved by using a smaller subset of features, solvation energies for the ligand or receptor. The results suggest that the proposed approach also performs feature selection, a problem that is currently being investigated. Other interesting problems that deserve investigation are the use of this approach in different protein-protein interaction classification problems, including intra and inter domains, homo and hetero-oligomers, and the use of other features, such as solvent accessibility, residual vicinity, shape of the structure of the interface, secondary structure, planarity, conservation scores, physicochemical features, hydrophobicity and others.

Acknowledgments

This research work has been partially supported by NSERC, the Natural Sciences and Research Council of Canada, grant No. RGPIN 261360, and the University of Windsor, internal Start-up grant.

References

1. Janin, J.: Kinetics and thermodynamics of protein-protein interactions from a structural perpective. In: Protein-Protein Recognition, p. 344. Oxford University Press, Oxford (2000)
2. Jones, S., Thornton, J.M.: Analysis and classification of protein-protein interactions from a structural perspective. In: Protein-Protein Recognition. Oxford University Press, Oxford (2000)
3. Russell, R.B., Alber, F., Aloy, P., Davis, F.P., Korkin, D., Pichaud, M., Topf, M., Sali, A.: A structural perspective on protein-protein interactions. Curr. Opin. Struct. Biol. 14(3), 313–324 (2004)
4. Kurareva, I., Abagyan, R.: Predicting molecular interactions in structural proteomics. In: Nussinov, R., Shreiber, G. (eds.) Computational Protein-Protein Interactions, pp. 185–209. CRC Press, Boca Raton (2009)
5. Ofran, Y.: Prediction of protein interaction sites. In: Nussinov, R., Shreiber, G. (eds.) Computational Protein-Protein Interactions, pp. 167–184. CRC Press, Boca Raton (2009)
6. Lawrence, M.C., Colman, P.M.: Shape complementarity at protein/protein interfaces. J. Mol. Biol. 234(4), 946–950 (1993)
7. Chakrabarti, P., Janin, J.: Dissecting protein-protein recognition sites. Proteins 47(3), 334–343 (2002)
8. Gnatt, A.L., Cramer, P., Fu, J., Bushnell, D.A., Kornberg, R.D.: Structural basis of transcription: an RNA polymerase II elongation complex at 3. 3 A resolution. Science 292(5523), 1876–1882 (2001)
9. Xu, D., Tsai, C., Nussinov, R.: Hydrogen bonds and salt bridges accross protein-protein interfaces. Protein Eng. 10(9), 999–1012 (1997)
10. Shanahan, H., Thornton, J.: Amino acid architecture and the distribution of polar atoms on the surfaces of proteins. Biopolymers 78(6), 318–328 (2005)
11. Ma, B., Elkayam, T., Wolfson, H.: Protein-protein interactions: structurally conserved residues distinguish between binding sites and exposed protein surfaces. Proc. Natl. Acad. Sci., USA 100(10), 5772–5777 (2003)
12. Ofran, Y., Rost, B.: Analysing six types of protein-protein interfaces. J. Mol. Biol. 325(2), 377–387 (2003)
13. Nooren, I., Thornton, J.: Diversity of protein-protein interactions. EMBO Journal 22(14), 3846–3892 (2003)
14. Jones, S., Thornton, J.M.: Principles of protein-protein interactions. Proc. Natl Acad. Sci, USA 93(1), 13–20 (1996)
15. Glaser, F., Steinberg, D.M., Vakser, I.A., Ben-Tal, N.: Residue frequencies and pairing preferences at protein-protein interfaces. Proteins 43(2), 89–102 (2001)
16. Zhu, H., Domingues, F., Sommer, I., Lengauer, T.: Noxclass: Prediction of protein-protein interaction types. BMC Bioinformatics 7(27) (2006) doi:10.1186/1471–2105–7–27
17. Young, J.: A role for surface hydrophobicity in protein protein recognition. Protein Sci. 3, 717–729 (1994)
18. LoConte, L., Chothia, C., Janin, J.: The atomic structure of protein-protein recognition sites. J. Mol. Biol. 285(5), 2177–2198 (1999)
19. Qi, Y., Bar-Joseph, Z., Klein-Seetharaman, J.: Evaluation of different biological data and computational classification methods for use in protein interaction prediction. Proteins 63(3), 490–500 (2006)

20. Bordner, A.J., Abagyan, R.: Statistical analysis and prediction of protein-protein interfaces. Proteins 60(3), 353–366 (2005)
21. Caffrey, H.J., Somaroo, S.: Are protein protein interfaces more conserved in sequence than the rest of the protein surface? Protein Science 13, 190–202 (2004)
22. Neuvirth, S., Raz, R.: ProMate. a structure based prediction program to identify the location of protein protein binding sites. J. Mol. Biol. 338, 181–199 (2004)
23. Zhou, H., Shan, Y.: Prediction of protein interaction sites from sequence profile and residue neighbor list. Proteins 44(3), 336–343 (2001)
24. Park, S.H., Reyes, J., Gilbert, D., Kim, J.W., Kim, S.: Prediction of protein-protein interaction types using association rule based classification. BMC Bioinformatics 10(36) (2009) doi:10.1186/1471-2105-10-36
25. Mintseris, J., Weng, Z.: Structure, function, and evolution of transient and obligate protein-protein interactions. Proc. Natl. Acad. Sci., USA 102(31), 10930–10935 (2005)
26. Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I., Bourne, P.: The Protein Data Bank. Nucleic Acids Research 28, 235–242 (2000)
27. Rueda, L., Herrera, M.: Linear Dimensionality Reduction by Maximizing the Chernoff Distance in the Transformed Space. Pattern Recognition 41(10), 3138–3152 (2008)
28. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. John Wiley and Sons, Inc., New York (2000)
29. Fisher, R.: The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics 7, 179–188 (1936)
30. Loog, M., Duin, P.: Linear Dimensionality Reduction via a Heteroscedastic Extension of LDA: The Chernoff Criterion. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(6), 732–739 (2004)
31. Ivanciuc, O.: Applications of Support Vector Machines in Chemistry. In: Reviews in Computational Chemistry, pp. 291–400. Wiley, Chichester (2007)
32. Camacho, C., Zhang, C.: FastContact: rapid estimate of contact and binding free energies. Bioinformatics 21(10), 2534–2536 (2005)
33. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 3rd edn. Elsevier Academic Press, Amsterdam (2006)

Employing Publically Available Biological Expert Knowledge from Protein-Protein Interaction Information

Kristine A. Pattin, Jiang Gui, and Jason H. Moore

Dartmouth Medical School, Lebanon, NH 03756, USA
kristine.a.pattin@dartmouth.edu
www.epistasis.org

Abstract. Genome wide association studies (GWAS) are now allowing researchers to probe the depths of common complex human diseases, yet few have identified single sequence variants that confer disease susceptibility. As hypothesized, this is due the fact that multiple interacting factors influence clinical endpoint. Given the number of single nucleotide polymorphisms (SNPs) combinations grows exponentially with the number of SNPs being analyzed, computational methods designed to detect these interactions in smaller datasets are thus not applicable. Providing statistical expert knowledge has exhibited an improvement in their performance, and we believe biological expert knowledge to be as capable. Since one of the strongest demonstrations of the functional relationship between genes is protein-protein interactions, we present a method that exploits this information in genetic analyses. This study provides a step towards utilizing expert knowledge derived from public biological sources to assist computational intelligence algorithms in the search for epistasis.

Keywords: GWAS, SNPs, Protien-protein interaction, Epistasis.

1 Introduction: Challenges Confronting Genome-Wide Genetic Analysis

The field of human genetics is advancing with the advent of new and cost efficient technology that allows us to rapidly generate large amounts of genomic data. It is now possible to measure a million or more SNPs at one time, however researchers are lacking methods to efficiently explore their results. The etiology of common human disease is understood to be complex, with multiple interacting genetic factors predisposing individuals to disease risk. To detect and characterize these epistatic, or gene-gene, interactions that confer disease susceptibility in such large scale studies requires the analysis of all pair-wise and higher-order combinations of SNPs. This poses a challenge that needs to be addressed before we are able to completely explore epistasis in GWAS in order to gain a more coherent understanding of the genetic architecture of a complex trait and its interacting elements [1].

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 395–406, 2010.

The critical need for statistical and computational methods that are powerful enough to model the relationship between SNP interactions and disease susceptibility has been addressed by the development of numerous statistical, machine learning, and datamining techniques. Some programs of note are multifactor dimensionality reduction (MDR) [2], ReliefF [3] and random chemistry [4]. Even though these methods have proven to be an effective way to model epistasis in smaller datasets, analysis of all higher-order SNP combinations in GWAS remains computationally infeasible.

One approach to this problem is to use computational intelligence algorithms that are able to explore a fitness landscape that is both vast and rugged. An important feature of this problem domain is that it is often the case that the attributes to be detected and modeled don't have detectable marginal effects and thus don't make good building blocks [5]. The use of statistical expert knowledge has been shown to improve the ability of learning algorithm to identify and exploit those building blocks that will yield an informative classifier Greene et al. (2008), have previously combined the power of ant colony optimization as a probabilistic learner with expert knowledge in the form of preprocessed ReliefF scores that reflect attribute quality and thus provide a measure of whether a particular attribute is a good building block [6][7]. Also Greene et al., (2009) and Moore et al., (2008) have shown that the same statistical expert knowledge incorporated in a computational evolution system (CES), has the same beneficial effect. They demonstrated that the system could learn to recognize and exploit a good source of expert knowledge from among several different options to discover optimal solutions in this problem domain [8],
cite2greene09. While preprocessed statistical knowledge is useful, it is likely not comprehensive in its ability to identify good building blocks. We anticipate that biological knowledge derived from biochemical pathways or regulatory networks of function, such as protein-protein interactions, will provide the complementary information that is needed to maximize the ability of a computational intelligence algorithm to identify optimal models of epistasis [10][11]. The goal of the present study is to explore the bioinformatics methods that are necessary to extract and utilize expert knowledge from public protein-protein interaction databases.

For this study we develop metrics derived from PPI interactions found in the database, STRING (Search Tool for the Retrieval of Interacting Genes/Proteins). These are based on the confidence score for each interaction in the database [12] and used to to prioritize SNPs in a gene list that is derived from a real bladder cancer dataset. While reducing the size of a genomic dataset using this approach may be useful to conduct a more computationally efficient analysis, we wish to ultimately explore how this protein interaction information can be used to guide a computational intelligence algorithm.

2 Materials and Methods

2.1 STRING

This study aims to understand how we can utilize expert knowledge from protein-proteins interactions to guide the search for epistasis in GWAS. We extract our biological expert knowledge from STRING (Search Tool for the Retrieval of Interacting Genes/Proteins). The newest version of STRING, 8.2, represents over 2.5 million proteins from 630 different organisms, and incorporates PPI information from a number of interaction databases such as the Human Protein Reference Database (HPRD) , BioGrid, the Molecular Interaction Database (MINT), the Biomolecular Interaction Network Database (BIND) which is a component of the Biomolecular Object Network Database(BOND), the Database of Interacting Proteins (DIP), and also imports known reactions from Reactome and Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways. Recent additions to this database incorporate information from IntACT, EcoCyc, NCI-Nature Pathway Interaction Database and the Gene Ontology (GO). Automated text-mining of PubMed abstracts, Online Mendelian Inheritance in Man (OMIM), and data from other databases such as the Saccharomyces Genome Database, Wormbase, and the Interactive Fly supplement this information [13]. At the time of our study however, we utilized the then current version available for download, version 7.0. Each interaction has a combined confidence score that ranges from 0 to 1 and are based on each source of evidence. Computing the combined confidence score, which we use as our source of expert knowledge, is a simple expression of the individual scores for each source of evidence. For a detailed description of the scoring methods see (von Mering et al., 2005) [12].

2.2 Interaction Scenario Simulation

In order to develop and test our method, we identified 8 different gene-gene interaction scenarios present in a list of genes derived from a real genetic bladder cancer dataset. The purpose of this was to represent a range of different gene-gene interaction scenarios that had validated interactions in STRING so that we may determine how PPI interaction information can be used as a source of expert knowledge should these biological interactions represent actual statistical epistatic interactions. Therefore, these interactions represent only theoretical statistical interactions. The original genetic data were originally collected for the purpose of assessing genetic risk factors of bladder cancer (Andrew et al., 2008). This dataset genotyped 491 cases and 791 controls across 1,423 SNPs found in 394 genes [14].

The gene-gene interactions designated in each scenario were selected to represent a range of combined confidence scores. As described, von Mering et al.,(2005) have developed the scoring system for PPIs in STRING and consider a score <0.7 and >0.4 as a medium confidence interaction [12]. The default threshold for querying interactions in STRING is 0.4, and we kept this setting when submitting the entire list of genes in our bladder cancer dataset to STRING. We chose interactions that exhibited a range of confidence for each interaction: BARD1-BRCA1 (confidence = 0.999), CASP3-CASP9 (confidence = 0.999), IL4R-IL6R (confidence = 0.825), RET-ENG (confidence = 0.7), TERT-MTR (confidence = 0.532), HSD3B1-SOD1 (confidence = 0.497), and two pairs that showed no existence of interaction in STRING, DRD4-BIC and RERG-SCUBE2 [Table 1].

Table 1. PPI interaction pairs represented in our bladder cancer data. The combined confidence score from STRING for each interaction and the number of interactors for each gene. Each of the interaction pairs was chosen to represent a range of interaction scores and diverse interaction scenarios.

PPI	Confidence Score	# of Interactors ≥ 0.4
CASP9 CASP3	0.999	29 58
BARD1 BRCA1	0.999	17 71
IL6R IL4R	0.825	24 38
ENG RET	0.7	12 13
MTR TERT	0.532	7 18
SOD1 HSD3B1	0.497	15 8
BIC DRD4	None	0 0
SCUBE2 RERG	None	0 0

Note that if any disease or pathway relationship exists between these genes it is by chance, given that genes were not selected on this basis.

2.3 Metrics

Expert knowledge from protein-protein interactions was employed to develop metrics that we used prioritize genes in a genomic dataset. The goal was to determine if PPIs will provide a valuable source of expert knowledge to preprocess data and eventually evaluate the effectiveness of this approach in a computational intelligence algorithm. The combined confidence score for protein-protein interactions from the STRING database was used to prioritize the genes a bladder cancer gene list according to each metric we developed. Note that all SNPs represented in the bladder cancer dataset that are found in the same gene are assigned the same metric score and rank using this method.

First, all 394 genes in the bladder cancer dataset were queried to obtain the list of combined confidence scores for all existing interactions between genes that represent PPIs in STRING. Using this combined confidence score, metrics were developed as follows:

For gene X that interacts with N number of genes with descending confidence scores $n_1 \ldots n_i$ we computed,

$$Average(\text{AVE}),$$

$$\text{AVE}\,X = \frac{\sum(n_1 \ldots n_i)}{N}, \tag{1}$$

the average of all confidence scores for a gene and its interaction partners, and

$$Sum(\text{SUM}),$$

$$\text{SUM}\,X = \sum(n_1 \dots n_i), \tag{2}$$

the sum of all confidence scores for a gene and its interaction partners.

While the maximum confidence score was evaluated as a metric, it was discovered that a considerable number of genes are represented by the same maximum confidence score. This was not useful for the individual prioritization of all genes, thus two additional metrics, MAX-SUM and MAX-AVE were developed. In this case, the metric score for each gene is represented by the numerical rank for that gene in the dataset after prioritization. These were developed as follows:

For gene $X_1 \dots X_i$, those with the same maximum confidence score are sorted in descending order by their SUM X or by AVE X.

The MAX-SUM metric becomes the numerical rank of the gene as they are prioritized in descending order.

To determine if using a specific cut-off of 0.4 had any effect on the metric scores for each gene, we recalculated each metric using lower thresholds of 0.2 and 0.3, and higher thresholds of 0.5, 0.6, 0.7, and 0.8. Metrics calculated using these thresholds were compared with each other and also to our original metrics. The comparison was performed using the Wilcoxon rank sum test as described in the next section, however, significant difference was not observed after implementing these threshold cutoffs (data not shown).

2.4 Evaluating Metrics

To evaluate these metrics, genes in the bladder cancer dataset were prioritized by their metric score and evaluated by extent by which each metric could reduce the entire gene list while retaining each of the 8 validated interaction pairs. This included two scenarios where genes had no evidence of interacting in STRING. Genes were sorted in descending order after being assigned their individual metric scores. The numerical position of each gene within the prioritized list represents the rank for that gene and is expressed in terms of the number of SNPs included in the gene. Since the actual bladder cancer dataset is comprised of SNP data, all SNPs in one gene share this rank. This means that genes with a higher metric score would have a smaller numerical rank. For example, the highest scoring gene in the list would be 1, but may have 5 SNPs within that gene, therefore making the rank for this gene, 5.

The gene list was then truncated at the lowest scoring gene in each interaction pair so that both of these genes, inclusive of all their SNPs, were maintained in the truncated list. For example, expressing the size of this list in terms of SNPs, applying the AVE metric to the CASP3 - CASP9 scenario, if we truncated the list at CASP3, this would include 315 SNPs, (315 being considered its rank) but this would omit its interaction partner CASP9 since it had a larger numerical rank. To include CASP9, the list had to be truncated at 599 SNPs. This larger numerical rank for each interaction pair was the number we used in order to compare the metrics, not the metric score itself.

The ability of each metric to significantly reduce the gene list in each interaction scenario was evaluated using a pair-wise one sided Wilcoxon rank sum test to compare the ranks obtained by each metric.A p-value of < 0.05 was considered to be indicate significant reduction. The non-interacting SNPs were not included in this comparison, because it was confirmed that for all metrics, their rank was 0, and for both pairs to be included in the gene list , all 1423 SNPs had to be considered. This was expected since these scenarios served as control to assure that our computation and ranking system were functioning appropriately.

We also examine whether there was a correlation between the confidence score for a given interaction and the ranking for the pairs across all metrics. To evaluate this we performed a Spearman Rank correlation test and considered there to be a significant correlation for p-values < 0.05.

2.5 Bladder Cancer Data

Finally, metrics were examined in the context of the actual gene-gene interactions that were indentified in a variation of the bladder cancer dataset utilized throughout this study. Andrew et al. (2008) used a multifactor analysis strategy to investigate associations between DNA repair polymorphisms and bladder cancer risk. Gene-environment and gene-gene interactions were evaluated using logistic regression, MDR, hierarchical interaction graphs, classification and regression trees, and logic regression analyses. All methods supported an interaction between DNA repair polymorphisms XRCC1-399/XRCC3-241 ($p = 0.001$), and three methods identified an interaction between XRCC1-399/XPD-751 ($p = 0.008$). The ranks of these gene pairs using each of our metrics were calculated to determine if they would have been effective in this real genetic analysis.

3 Results

In total there were 357 interacting proteins within the dataset and a total of 3,921 different interactions amongst these proteins. A full list of all interactions and their individual confidence score and the metric score for each individual gene are available upon request.

Figure 1 demonstrates the extent to which each metric was able to reduce the size of the gene list for each of the 6 interaction scenarios. The confidence score for each gene pair interaction is represented on the x-axis in ascending order, and the ranking for each of these is represented on the y-axis. This ranking represents the extent to which the gene list could be reduced by applying each metric. Table 2 displays the individual rankings produced for each scenario after being prioritized by each metric. In a vast majority of the cases, higher confidence interactions are found in smaller subsets of the gene list when all metrics are applied, however this is not a consistent trend. For example, even though HSD3B1-SOD1 has the lowest confidence score, it is found in a smaller subset of the list than RET-ENG (confidence=0.7) when using AVE, MAX-SUM, and MAX-AVE. Both SUM and

AVE produce larger subsets of the gene list, except for in the case of the RET-ENG scenario for both and also in the case of CASP9-CASP3 for SUM. The MAX-SUM metric produces similar ranks as MAX-AVE, yet these decrease as confidence score increases.

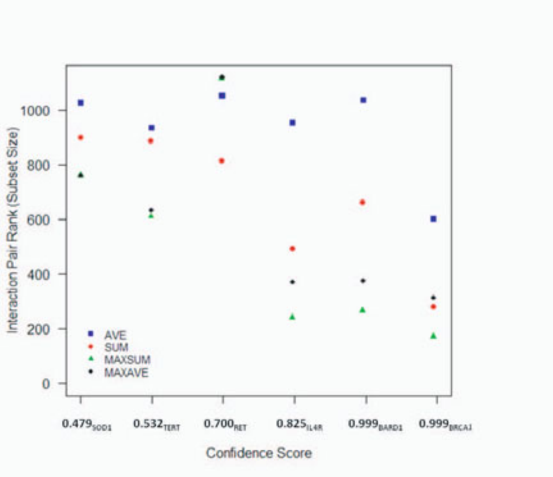


Fig. 1. Depicts the extent to which each metric was able to reduce the gene list for each of the 6 functional interactions. We did not include the non interacting pairs. The confidence score for each SNP pair interaction is represented on the x-axis in ascending order, and the ranking for each interacting SNP pair is represented on the y-axis. The rank is also indicative of the subset size to which the gene list was reduced.

To determine if these observations were significant, a one sided pair-wise Wilcoxon rank sum test was used to compare each metric in terms of the gene rankings they produced across the interaction scenarios [Table(3)]. The metric AVE produced significantly larger ranks than SUM, MAX-SUM, and MAX-AVE (p= 0.017, 0.031, 0.031). This means that compared to these 3 metrics, AVE was not able to reduce the gene list as effectively as the others while retaining the interaction pairs. SUM showed no significant difference from MAX-SUM, or MAX-AVE, yet as mentioned, produced smaller rankings than AVE. MAX-SUM, however produced the lowest mean rank amongst the metrics (525) and also had significantly lower ranks than AVE and MAX-AVE (p=0.031 and p=0.030). When examining the individual numerical pair ranks produced by SUM and MAX-SUM, it is clear that all the ranks for MAX-SUM are much smaller than SUM except in the case of the rank of RET-ENG. We determined this to be the cause of the lack of significant difference between the two metrics. Note that SUM showed no significant difference compared to MAX-AVE either on account of the smaller rank score it produced for RET-ENG. Overall, MAX-SUM was able reduce the size of the gene list to the greatest extent in a majority of the interaction scenarios as compared to the other metrics despite the lack of significant difference between this metric and SUM.

Table 2. Shows the ranking for each interaction pair which is also the subset size of the gene list after genes were prioritized by each metric. The non-interacting SNPs do not have any interactors so therefore will only be included in the gene list as a whole since they cannot be ranked by the metrics.

PPIs	AVE	SUM	MAX-AVE	MAX-SUM
CASP3.CASP9	599	279	313	169
BARD1.BRCA1	1032	661	374	264
IL4R.IL6R	950	491	370	239
RET.ENG	1048	811	1118	1112
TERT.MTR	931	885	632	608
HSD3B1.SOD1	1022	898	758	758
DRD4.BIC	1423	1423	1423	1423
RERG.SCUBE2	1423	1423	1423	1423

Examining the relationship between the confidence of an interaction and its rank, it's observed that while an interaction with a higher confidence score may have a lower rank, this was not the case for all scenarios. However, there was a significant relationship between the two by means of Spearman Rank Correlation test. This showed that there was a negative correlation between the numerical rankings produced by each metric ($p < 0.05$), except AVE. Rank increased as the confidence score of the interacting pairs decreased [[Table(3)].

Table 3. Shows the metrics as compared to each other by one-sided pair-wise Wilcoxon rank sum test. Significance is indicated by (*) and directionality listed below.
*AVE > SUM, MAXSUM, MAXAVE
*MAXSUM < AVE, MAXAVE
*MAXAVE < AVE

.	AVE	SUM	MAX-SUM	MAX-AVE	Spear. P-Val.	Corr. Coeff.
AVE	.	.	.	.	0.40	-0.15
SUM	0.016*	.	.	.	0.007*	-0.9
MAX-SUM	0.031*	0.160	.	.	0.05*	-0.72
MAX-AVE	0.031*	0.290	0.030*	.	0.05*	-0.72

We applied our metrics to the bladder cancer dataset in the context of the interactions identified by Andrew et al. (2008). These results are described in [Table (4)]. Both interactions, XRCC1-XPD and XRCC1-XRCC3, were supported in STRING with confidence scores of 0.930 and 0.774, respectively. We find that by applying the SUM and AVE metrics in both cases, the dataset could be reduced to between 288 and 346 SNPs. However unlike in the simulated interaction scenarios, MAX-AVE and MAX-SUM produced larger ranks for these pairs and would have only been able to reduce the dataset to between 1126 and 1158 SNPs from the original 1423 SNPs represented.

Table 4. Shows the pair ranking for the gene-gene interactions identified by Andrew et al. (2008) when each metric was applied to their bladder cancer data. The interactions identified were between SNPs in each of the DNA repair genes listed.

DNA Repair Gene Interactions	AVE	SUM	MAX-AVE	MAX-SUM
XRCC1.XRCC3	288	346	1126	1176
XRCC1.XPD	288	234	1130	1158

4 Discussion

The goal of our study was to develop metrics based on protein-protein interaction information that would allow us to prioritize the SNPs in genomic datasets. We consider this a first step towards understanding how we can utilize expert knowledge derived from public biological sources, such as PPI databases, to facilitate the search for epistasis in GWAS. We use the combined confidence score for PPIs in STRING to develop these metrics and applied them to a real bladder cancer dataset from which we derived different gene-gene interaction scenarios. We found that of the four metrics that we have developed (AVE, SUM, MAX-SUM, and MAX-AVE), MAX-SUM was able to reduce the gene list in our datasets to the greatest extent across a majority of the interaction scenarios. While we did see a negative correlation between ranks and confidence score, except when AVE was used, [Table(3)]with higher confidence interactions typically being found in smaller subsets of the gene list, we recognize that the nature of the interaction may greatly influence this.

This was further supported by what we observed when we applied our metrics to the real interaction scenarios identified by Andrew et al. (2008). In our simulated scenarios, SUM and AVE produced larger subset sizes than MAX-SUM and MAX-AVE. However, it appears as though SUM and AVE would have been ideal to apply to the bladder cancer dataset in terms of the real statistical interactions that were observed. These metrics were able to reduce the dataset to a size that was approximately four times smaller in both cases, and despite the different confidence level of the interactions, 0.774 and 0.993. While this demonstrates that one metric may not be particularly more useful over the others, we show that expert knowledge from PPIs could have been applied to this study to narrow the scope of the analysis and still obtain the results that were previously published.

The diversity of the interaction scenarios [Table (1)] is why we do not see one particular metric consistently reducing the gene list, and also why interaction scenarios of higher confidence are not always included in the smaller subsets of the gene list, such as in the case of RET-ENG.

For the RET-ENG scenario, we see that the pair is included in the largest subsets produced by all metrics except SUM, even though it has a confidence score of 0.7. We find that RET interacts with 13 proteins and ENG with 12. However, out of these interactions, there are few high confidence interactions, especially for RET. More than half of its interactions have a confidence score under 0.7. Also, RET has a lower maximum confidence score of 0.874 which automatically ranks it as 1118 on the entire gene list inclusive of all SNPs. As mentioned, since the pair ranks are based on the gene with the higher numerical rank in the pair, we see this pair included in larger subsets due to the lack of a higher maximum score for RET using MAX-SUM and MAX-AVE. While the AVE score for its interactions might also be fairly low, it does have a considerable number of interaction partners, allowing for it to achieve a high metric score using SUM.

These examples and our application to the real statistical interactions exhibit why we chose not to consider one metric as the "best" metric, and acknowledge that there is room for exploration concerning how the combined protein interaction score from STRING can be utilized as biological expert knowledge. All metrics have the potential provide useful information to genetic studies, and we have shown that two of our metrics would have been able to effectively reduce the bladder cancer dataset while retaining the epistatic interactions that were previously shown to confer disease risk.

5 Conclusion

To exhaustively search thousands to millions of SNP combinations would not be practical due to the computational intensity involved in the process, and our work demonstrates one method that can be used to facilitate this search. The use of expert knowledge for these purposes is not a novel idea and has proven successful in similar endeavors [15][16][17][18]. However, there are potential drawbacks to using information from PPIs that should be acknowledged.

Certainly, there are inherent biases in any protein interaction database. This is something that is unavoidable and partially results from the ability of different experimental methods to capture different types of interactions. Also, a handful of proteins may be more widely studied. These facts influence the amount of evidence, whether it be via literature or other sources, that in turn determine the confidence placed in an interaction. Also, some may argue that by using expert knowledge, we are biasing a study, instilling the need for a prior hypothesis. While this may be valid, regardless we still do not have the appropriate computational power to explore higher order epistatic interactions in genome-wide datasets. By acknowledging the potential database biases, a researcher has the power to take these biases into consideration when conducting their analysis and understand how, if at all, this may have influenced their results.

We have shown that the protein interaction confidence score in the STRING database can be represented in a number of ways that may indicate the validity of an interaction as well as how central a gene is to the dataset based on the number of interactors it has within that dataset. If this holds true and there is biological representation of statistical results in an analysis, the application of our methods for prioritizing SNPs and reducing the scope of the analysis has a higher probability of retaining functional epistatic interactions after the data is processed. The development of computational methods that aid in the discovery and characterization of epistatic interactions in GWAS is of great importance, and this study opens the door for the utilization of expert knowledge from PPIs to guide these methods.

While these metrics are useful for preprocessing GWAS data, it is possible we may be removing potentially important information by truncating a dataset. To this end, we additionally would like to explore how to use this form of biological expert knowledge in the computational evolution system we have developed (CES), that has the ability to identify complex disease-causing genetic architectures in simulated data by exploiting valid sources of statistical expert knowledge. Unlike preprocessed data, all SNPs in the dataset would be considered in this case. We anticipate that this biological expert knowledge will improve the performance of this system.

Ultimately, we hope that the use of this biological expert knowledge will provide a complement to the statistical expert knowledge that we have already shown to be successful in such a system. This can supplement an analysis with a foundation for interpreting epistatic models and understanding how they influence disease risk biologically.

References

1. Moore, J., Ritchie, M.: The challenges of whole-genome approaches to common diseases. JAMA 29, 1642–1643 (2004)
2. Ritchie, M., Hahn, L., Roodi, N., et al.: Multifactor dimensionality reduction reveals high-order interactions among estrogen metabolism genes in sporadic breast cancer. Am. J. Hum. Genet. 69, 138–147 (2001)
3. Moore, J., White, B.: Tuning reliefF for genome-wide genetic analysis. In: Marchiori, E., Moore, J.H., Rajapakse, J.C. (eds.) EvoBIO 2007. LNCS, vol. 4447, pp. 166–175. Springer, Heidelberg (2007)
4. Eppstein, M., Payne, J., White, B., Moore, J.: Genomic mining for complex disease traits with 'random chemistry'. Genetic Programming and Evolvable Machines 8, 395–411 (2007)
5. White, B., Gilbert, J., Reif, D., et al.: A statistical comparison of grammatical evolution strategies in the domain of human genetics. In: Proceedings of the IEEE Congress on Evolutionary Computing, pp. 676–682 (2005)
6. Greene, C., White, B., Moore, J.: Ant colony optimization for genome-wide genetic analysis. In: Dorigo, M., Birattari, M., Blum, C., Clerc, M., Stützle, T., Winfield, A.F.T. (eds.) ANTS 2008. LNCS, vol. 5217, pp. 37–47. Springer, Heidelberg (2008)
7. Greene, C., Gilmore, J., Kiralis, J., et al.: Optimal use of expert knowledge in ant colony optimization for the analysis of epistasis in human disease. In: Pizzuti, C., Ritchie, M.D., Giacobini, M. (eds.) EvoBIO 2009. LNCS, vol. 5483, pp. 92–103. Springer, Heidelberg (2009)
8. Moore, J., Andrews, P., Barney, N., et al.: Development and evaluation of an open-ended computational evolution system for the genetic analysis of susceptibility to common human diseases. In: Marchiori, E., Moore, J.H. (eds.) EvoBIO 2008. LNCS, vol. 4973, pp. 129–140. Springer, Heidelberg (2008)
9. Greene, C., Hill, D., Moore, J.: Environmental sensing of expert knowledge in a computational evolution system for complex problem solving in human genetics. In: Riolo, R., O-Reilly, U.-M., McConaghy, T. (eds.) Genetic Programming Theory and Practice, vol. VII, pp. 19–36. Springer, Heidelberg (2009)
10. Pattin, K., Moore, J.: Exploiting the proteome to improve the genome-wide genetic analysis of epistasis in common human diseases. Hum. Genet. 124, 297–312 (2009)

11. Pattin, K., Moore, J.: Role for protein-protein interaction databases in human genetics. Exp. Rev. Proteomics 6, 647–659 (2009)
12. von Mering, C., Jensen, L., Snel, B., et al.: STRING: known and predicted protein-protein associations, integrated and transferred across organisms. Nucleic Acids Res. 1(33), D433–D437 (2005)
13. Jensen, L., Kuhn, M., Stark, M., et al.: STRING 8–a global view on proteins and their functional interactions in 630 organisms. Nucleic Acids Res. 37, D412–D416 (2009)
14. Andrew, A., Karagas, M., Nelson, H., et al.: Assessment of multiple DNA repair gene polymorphisms and bladder cancer susceptibility in a joint Italian and U.S. population: a comparison of alternative analytic approaches. Hum. Hered. 65, 105–118 (2008)
15. Emily, M., Mailund, T., Hain, J., et al.: Using biological networks to search for interacting loci in genome-wide association studies. Eur. J. Hum. Genet. 17(10), 1231–1240 (2009)
16. Bush, W., Dudek, S., Ritchie, M.: Biofilter: a knowledge-integration system for the multi-locus analysis of genome-wide association studies. In: Pac. Symp. Biocomput., pp. 368–379 (2009)
17. Shriner, D., Tesfaye, B., Padilla, M., et al.: Commonality of functional annotation: a method for prioritization of candidate genes from genome-wide linkage studies. Nucleic Acids Res. 36(4), e26 (2008)
18. Saccone, S., Saccone, N., Swan, G., et al.: Systematic biological prioritization after a genome-wide association study: an application to nicotine dependence. Bioinformatics 24, 1805–1811 (2008)

SFFS-MR: A Floating Search Strategy for GRNs Inference

Fabrício M. Lopes[1,2], David C. Martins-Jr[3],
Junior Barrera[4], and Roberto M. Cesar-Jr[2,5]

[1] Federal University of Technology - Paraná, Brazil
fabricio@utfpr.edu.br
[2] Institute of Mathematics and Statistics, University of São Paulo, Brazil
fabriciolopes,cesar{@vision.ime.usp.br}
[3] Federal University of ABC, Brazil
david.martins@ufabc.edu.br
[4] Faculty of Philosophy, Sciences and Letters of Ribeirão Preto,
University of São Paulo, Brazil
jb@ime.usp.br
[5] CTBE, Brazil

Abstract. An important problem in the bioinformatics field is the inference of gene regulatory networks (GRN) from temporal expression profiles. In general, the main limitations faced by GRN inference methods is the small number of samples with huge dimensionalities and the noisy nature of the expression measurements. In face of these limitations, alternatives are needed to get better accuracy on the GRNs inference problem. In this context, this work addresses this problem by presenting an alternative feature selection method that applies prior knowledge on its search strategy, called SFFS-MR. The proposed search strategy is based on SFFS algorithm, with the inclusion of multiple roots at the beginning of the search, which are defined by the best and worst single results of the SFS algorithm. In this way, the search space traversed is guided by these roots in order to find the predictor genes for a given target gene, specially to better identify genes presenting intrinsically multivariate prediction, without worsening the asymptotical computational cost of the SFFS. Experimental results show that the SFFS-MR provides a better inference accuracy than SFS and SFFS, maintaining a similar robustness of the SFS and SFFS methods. In addition, the SFFS-MR was able to achieve 60% of accuracy on network recovery after only 20 observations from a state space of size 2^{20}, thus presenting very good results.

Keywords: SFS, SFFS, feature selection, inference, gene networks, pattern recognition, systems biology, bioinformatics.

1 Introduction

Systems biology is the study of live organisms viewed as integrated and interacting networks of genes, proteins and biochemical reactions. It has been an emergent

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 407–418, 2010.

field of study in bioinformatics since the advent of high-throughput technologies for extraction of gene expressions (mRNA abundances or transcripts), such as DNA microarrays [29] or SAGE [35], and more recently RNA-Seq [36]. These high-throughput techniques together with computational methods, make possible to analyze thousands of transcripts simultaneously. Consequently, the volume of transcriptome data available from a multitude of species during the last ten years increased dramatically. In this context, a big challenge that researchers need to face is the large number of variables (thousands) for just a few experiments available (dozens). In order to infer relationships among those variables, it is needed a great effort in developing novel computational and statistical techniques that are able to alleviate the intrinsic error estimation committed in the presence of small number of samples with huge dimensionalities.

A commonly used approach to infer gene regulatory networks (GRN) is the use of feature selection techniques [18,4,13,23,21,26,11,3,38,9]. A feature selection method is composed by two main parts: a search algorithm and a criterion function. As far as the search algorithms are concerned, there are two main categories: the optimal and sub-optimal algorithms. The optimal algorithms (including exhaustive and branch-and-bound searches) return the best feature subspace, but their computational costs are very high to be applied in general, especially for high dimensionality problems such as GRN inference focused in this paper. The sub-optimal algorithms do not guarantee that the solution is optimal, but some of them present a reasonable cost-benefit between computational cost and quality of the solution. In this work, we explore the exhaustive search (optimal), the Sequential Forward Selection (SFS - sub-optimal) and the Sequential Forward Floating Selection (SFFS - sub-optimal with excellent cost-benefit) [24].

The reason why efficient search algorithms such as SFS and SFFS do not always reach the optimal solution is due to the *nesting effect* in which a feature not present in the optimal solution may be included in the partial solution of the algorithm and never be discarded, which leads to a sub-optimal solution [32]. Such effect can be explained by the fact that two features grouped in a pair may perform a very nice prediction of the class (or value) of the target object, although their individual predictions about the target are bad. Such pair of features can be even better than two other features grouped that perform well individually. This phenomenon is called synergy or intrinsically multivariate prediction [1,22].

The main contribution of this work is the proposal of a new search strategy algorithm that uses the very efficient SFFS starting from some initial features (denominated here as “roots”). Such roots contain not only the best individual features, but also the worst ones. This strategy tries to identify those features that are synergetic (or intrinsically multivariate predictive) in predicting the targets without worsening the asymptotical computational cost of the SFFS. The focus of the experiments presented here is given on the GRN inference application.

Next section (Section 2) will introduce a brief background on the network inference problem. In Section 3, the feature selection problem is discussed in more detail, including a short description of the SFS and SFFS techniques. Section 4 discusses the intrinsically multivariate prediction issue and how it can affect the greedy feature selection algorithms in such way that the achieved solution be relatively far away from the optimal. Section 5 describes our proposed feature selection method (SFFS-MR). Section 6 shows some experimental results. Finally, Section 7 concludes the work, discussing future perspectives.

2 GRN Inference

The inference of GRNs from temporal expression data is a great challenge. One of the main reasons for that is the usual limitation of the data itself in which the number of samples is not enough to infer relationships among elements present in the biological system with a reasonable confidence [15]. Due to this fact, there are several approaches proposed for modeling and identification of GRNs. The main approaches for modeling of GRNs are Boolean Networks, Differential Equations and Bayesian Networks. As Boolean Networks (and its stochastic version: Probabilistic Boolean Networks (PBN) [30]) are more suitable in situations with limited data samples, here we concentrate the attention in the PBN model.

Considering feature selection approaches for identification of GRNs, there are three types of criterion functions frequently used. The first one is the correlation between two features, in which there is an edge between two genes if the correlation between their expression profiles are larger than some predefined threshold [33]. Such method considers only 1-to-1 relationships, being suitable to identify co-regulation between genes, functional modules and clusters. Nevertheless, it ignores the fact that the expression of a given target gene may be regulated by a group of genes with multivariate interaction.

Another class of criterion functions refers to those based on the Bayesian error estimation of the predictors in classifying the target expression. A broadly used criterion to infer GRNs is the coefficient of non-linear determination (CoD) [13,7,12]. With this measurement, it is possible to capture N-to-1 (multivariate) relationships.

The information theory based (entropy, mutual information) criterion functions are also commonly employed for inference of GRNs. There are several works that use such metrics in substitution to correlation to infer 1-to-1 relationships in GRNs [4,25,21,11,26]. However, it is possible to employ these metrics for inference of multivariate relationships [18,3,38]. Basically, the difference between the entropy based and the Bayesian error based criteria is that the second relies on the minimum conditional probability distributions of the target given the subset of predictors, while the former relies on the uniformity of these conditional probability distributions as a whole (larger uniformity leads to higher entropy, which in turn leads to smaller mutual information).

The literature related to modeling and inference of GRNs is vast and continues to grow quickly, which reflects the importance of this research field. Some reviews on this topic can be found in [5,16,34,28,17,14].

3 Feature Selection

A feature selection method consists in selecting a subset of features that makes a good representation, classification or prediction of states (or values) of the objects in study. Generally, it is composed by two main parts: a search strategy and a criterion function that attributes a quality value to the feature subsets. Due to the nesting effect (see Section 4), the search for the best subsets generally requires the investigation of the entire space of possible subsets (exhaustive search), although depending on some constraints of the criterion function adopted (e.g. monotonical or U-shaped), it is possible to obtain the best subset by looking for a restricted space of subsets employing "branch-and-bound" techniques [31,27].

As exhaustive search is computationally impractical for most "real world" tasks, and especially for inference of GRNs which involves data with thousands of features (genes), it is clear the existence of a trade-off between optimality and computational cost. Next we introduce two classical heuristics for feature selection.

3.1 Sequential Forward Selection (SFS)

The SFS algorithm starts with the empty set and adds the best feature found to this set according to the criterion function adopted. In the next step, it adds a second feature that, jointly with the feature already included, composes the best feature pair. This process continues until it reaches a stop condition, commonly based on a fixed dimension (number of features of the subset to be returned), or based on the criterion function value variation (it stops if the criterion value does not improve significantly from the previous to the next step). There is a variant of this algorithm, the Sequential Backward Selection (SBS), which starts with the complete set and successively removes the less relevant features according to the criterion function until the stop condition be satisfied [24].

3.2 Sequential Forward Floating Selection (SFFS)

The SFS and SBS search methods present an undesirable drawback known as nesting effect. This effect happens because the discarded features in the top-down approach are not inserted anymore, or the inserted features in the bottom-up approach are never discarded. Section 4 presents the reason why this phenomenon occurs, and the potential problem leaded by it in some important gene inference situations.

In order to amenize this problem, the SFFS [24] was adopted. The SFFS algorithm tries to avoid the nesting effect by allowing to insert and to exclude features on a subset in a floating way, i.e., without defining the number of insertions or exclusions. In this algorithm, SFS and SBS are successively applied. A schematic flowchart of the SFFS algorithm is presented in Figure 1.

The aforementioned algorithm is computationally efficient and usually returns a solution very close to the optimal, presenting excellent cost-benefit. There are also adaptive and generalized floating methods that try to improve the SFFS

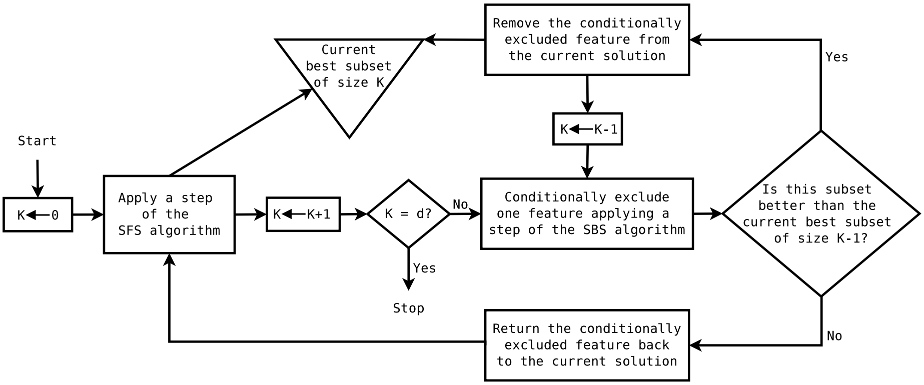

Start
K ← 0
Apply a step of the SFS algorithm
Current best subset of size K
K ← K+1
K = d?
No
Yes
Stop
Conditionally exclude one feature applying a step of the SBS algorithm
Remove the conditionally excluded feature from the current solution
K ← K-1
Is this subset better than the current best subset of size K-1?
Yes
No
Return the conditionally excluded feature back to the current solution

Fig. 1. Simplified flowchart of the SFFS algorithm [19]. K refers to the size of the current solution subset while d refers to the size of the final solution subset.

results at the expense of a significant increase on the computational cost. Nevertheless, they still can not avoid the nesting effect completely [32].

4 Intrinsically Multivariate Prediction

A set of predictor features is considered intrinsically multivariate predictive with regard to a target feature if the target behavior is strongly predicted by the whole set of predictors, but poorly predicted by any of its proper subsets. Formally, a set of features $\mathbf{X}$ is intrinsically multivariate predictive for the target feature Y with respect to λ and δ, for $0 \leq \lambda, \delta \leq 1$ and $\lambda < \delta$, if

$$\max_{\mathbf{Z} \subsetneq \mathbf{X}} \mathcal{F}_Y(\mathbf{Z}) \ \leq \ \lambda \wedge \mathcal{F}_Y(\mathbf{X}) \geq \delta \qquad (1)$$

where $\mathcal{F}$ is a criterion function that varies from 0 to 1 (0 meaning absence of prediction and 1 meaning full prediction) [22]. Generally, λ has small value (usually smaller than 0.2) and δ has a high value (usually larger than 0.6). For a pair predictors-target $(\mathbf{X}, Y)$, the largest δ for which the prediction is intrinsically multivariate is $\delta = \mathcal{F}_Y(\mathbf{X})$. In this way, it is possible to define a score of intrinsically multivariate prediction (IMP score) through the maximum value of $\delta - \lambda$. Thus, the IMP score is given by

$$I_Y(\mathbf{X}) = \mathcal{F}_Y(\mathbf{X}) - \max_{\mathbf{Z} \subsetneq \mathbf{X}} \mathcal{F}_Y(\mathbf{Z}) \qquad (2)$$

The concept of intrinsically multivariate prediction is related to the nesting effect that occurs when a greedy feature selection algorithm like SFS or other sub-optimal heuristics are applied. Next we present an example that clarifies this concept. Suppose two Boolean features $X_1 = x_1 \in \{0, 1\}$, $X_2 = x_2 \in \{0, 1\}$ and another Boolean feature $Y = y \in \{0, 1\}$ considered as target. Also suppose the joint probability distributions (JPD) $P(x_1, x_2, y)$ $\forall\{x_1, x_2, y\} \in \{0,1\}^3$ given in Table 1. Considering the nonlinear Coefficient of Determination (CoD) as criterion function defined as $CoD_Y(\mathbf{X}) = \frac{\varepsilon_Y - \varepsilon_Y(\mathbf{X})}{\varepsilon_Y}$, where ε_Y is the error obtained by classifying Y in the absence of other observations (prior error) and $\varepsilon_Y(\mathbf{X})$ is the error obtained by classifying Y based on the observation of the feature set $\mathbf{X}$ [8]. Such pair predictors-target has $CoD_Y(X_1, X_2) = \frac{0.5-0.2}{0.5} = 0.6$. On the other hand, if we consider X_1 and X_2 individually, both $CoD_Y(X_1)$ and $CoD_Y(X_2)$ are zero, since $P(X_1 = x_1, Y = y) = P(X_2 = x_2, Y = y) = 0.25$ $\forall x_1 \in \{0,1\}, x_2 \in \{0,1\}, y \in \{0,1\}$, which implies $CoD_Y(X_1) = CoD_Y(X_2) = \frac{0.5-0.5}{0.5} = 0$. The IMP score in this case is $I_Y(X_1, X_2) = 0.6 - 0 = 0.6$, which is considered high (X_1, X_2 and Y form an IMP set).

Table 1. Example of a joint probability distribution (JPD) between the target Y and two predictors X_1 and X_2 in which such features form an intrinsically multivariate predictive set ($I_Y(\mathbf{X}) = 0.6$ for CoD as criterion function).

$X_1 = x_1$	0	0	1	1
$X_2 = x_2$	0	1	0	1
$P(X_1 = x_1, X_2 = x_2, Y = 0)$	0.2	0.05	0.05	0.2
$P(X_1 = x_1, X_2 = x_2, Y = 1)$	0.05	0.2	0.2	0.05

It is important to notice that, in the example given above, Y is given by a stochastic exclusive-or (XOR), i.e., $argmax_{y \in Y} P(y|x_1, x_2) = 0$ if $x_1 = x_2$ or $argmax_{y \in Y} P(y|x_1, x_2) = 1$ if $x_1 \neq x_2$. According to [22], in the case of 2 binary predictors, there are 8 logics that can produce high IMP score: XOR, NXOR (negated XOR), AND, OR, NOR, NAND, $x_1 \wedge \bar{x_2}$ and $x_1 \vee \bar{x_2}$. However, there are other properties besides prediction logic that can jointly originate IMP sets: predictive power (defined as $1 - \varepsilon_Y(\mathbf{X})$), covariance between predictors and probability distribution of each isolated predictor (marginal probabilities).

Most feature selection heuristics discard features that perform bad individual prediction about the target to compose the initial solutions. Due to the intrinsically multivariate prediction phenomenon, such heuristics tend to reach local minima that sometimes are far from the optimal solution. Next section describes the main contribution of the paper: a feature selection strategy that applies SFFS starting not only looking for good features, but also searching for bad individual features which can form intrinsically multivariate predictive sets.

5 SFFS with Multiple Roots (SFFS-MR)

The inference of GRNs is one of the most challenging problems of Systems Biology in these days, mainly because of the intrinsic error estimation due to small number of samples with huge dimensionalities and the presence of genes that have an intrinsically multivariate prediction. In this context, we propose an alternative search strategy for GRNs inference problem, called SFFS-MR, which extends the SFFS method by including not only the best individual features, but also the worst ones. Algorithm 1 presents the specification of the SFFS-MR algorithm.

Algorithm 1. SFFS-MR ($\varDelta$, d, irb, irw)

```
var list $exelist$, $bestset$, $newsubset$
var vector $ibroots[irb]$, $iwroots[irw]$
var float $bestvalue$, $newvalue$
var integer $k \leftarrow 1$
$bestvalue \leftarrow$ SFS($ibroots$, $iwroots$, $k$)
$bestset \leftarrow ibroots[1]$
for $i = 1$ to $irb$ do
   $exelist$.append($ibroots[i]$)
end for
for $i = 1$ to $irw$ do
   $exelist$.append($iwroots[i]$)
end for
while $exelist$ is not empty do
   $newsubset \leftarrow exelist$.removefirst
   $k \leftarrow newsubset$.cardinality
   if $k < d$ then
      $newvalue \leftarrow$ SFS($newsubset$, $\emptyset$, $k + 1$)
      if $newvalue < bestvalue$ and $(bestvalue - newvalue) > \varDelta$ then
         $newvalue \leftarrow$ SBS($newsubset$)
         $bestvalue \leftarrow newvalue$
         $bestset \leftarrow newsubset$
      end if
      $exelist$.append($newsubset$)
   end if
end while
return $bestset$
```

The Algorithm 1 starts by applying the SFS (Section 3.1) in order to discover the *irb* best and *irw* worst individual features ($k = 1$), which are ranked according to the adopted criterion function. The variable *bestvalue* represents the criterion function value achieved by the best feature *bestset*[1]. These individual subsets are appended in an execution list (*exelist*). In the *while* loop, the first subset in the execution list will be removed and its cardinality will be tested. If its cardinality has not reached the limit, the SFS will increment its cardinality, i.e., to include a new feature that jointly with the features already present in *newsubset* composes the best feature subset with cardinality $k + 1$. The variable *newvalue* represents the criterion function value achieved by *newsubset*. If *newsubset* has a better criterion function value (lower or higher, depending on the adopted criterion function), and the gain is more than $\varDelta$, then a conditional exclusion is performed, which is represented by calling SBS function, and the *bestvalue* and *bestset* are updated by *newvalue* and *newsubset* respectively. At the end, *newsubset* will be stored in the execution list for a new attempt to extend. In summary, the SFFS-MR differs from SFFS (Section 3.2) because of the exploration of multiple roots. If irb and irw (number of initial roots) are small constant values compared to the total number of variables (order of thousands for the application focused here), its asymptotical computational cost is not worse than SFFS.

The parameter d represents the maximum cardinality of the subset of predictors. A Δ of criterion function value variation is also included. Here the Δ value prevents that minor variations of the criterion function ($\leq \Delta$) causes the increase of the subset of predictors. The present paper adopted $d = 5$, $\Delta = 0.05$, $irb = 1$ and $irw = 5$.

6 Experimental Results

This section presents the experimental results obtained by considering a synthetic networks approach, which was adapted from [20]. The artificial gene networks (AGNs) were generated by considering the uniformly-random Erdös-Rényi (ER) [10] topology. The Probabilistic Boolean Networks (PBN) [30] approach was applied to generate the network dynamics, i.e., the temporal expression profiles.

For all experiments, the network model (ER) was applied with 20 vertices (genes). The average degree $\langle k \rangle$ per gene varied from 1 to 5, and the number of observed instants of time (signal size) varied from 5, 10, 15, 20 to 100 in steps of 20. For each gene g_i of the network, its value was given by a randomly selected function from 3 possible Boolean functions $\{f_1^{(i)}, f_2^{(i)}, f_3^{(i)}\}$, where the probabilities of each function be selected are given by $c_1^{(i)} = 0.95,\ c_2^{(i)} = 0.025,\ c_3^{(i)} = 0.025,\ i = 1, \ldots, 20$.

In order to identify the networks, the simulated temporal expressions were submitted to the software described in [19] which implements feature selection methods for network inference, applying the SFS and SFFS as search strategies. The same method and parameters (default) were kept fixed during comparative analysis with SFFS-MR. We adopted as similarity measure between the AGN and the inferred network, the PPV (Positive Predictive Value, also known as accuracy or precision) and Sensitivity (or recall) measurements presented by [6], which are widely used to compare the results of the GRNs inference methods. The experimental results were obtained from 50 simulations of each signal size and $\langle k \rangle$ value.

The first experiment was performed in order to compare the three methods: SFS, SFFS and SFFS-MR with respect to the temporal expressions size. Fig. 2 presents these results, in which the PPV measure was calculated by taking into account the average results for all variations of average degree $\langle k \rangle$.

It is possible to notice that all methods have an increase on its performance by increasing the number of observations. However, the improvement of the SFFS-MR was more consistent, e.g., achieving 60% of similarity against 55%

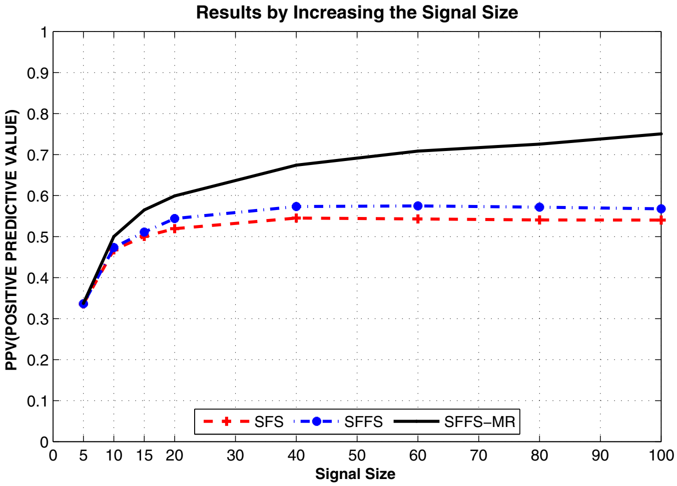


Fig. 2. PPV measure obtained by SFS, SFFS and SFFS-MR applied to infer network edges from different sizes of temporal expression profiles (signal size). Similarity measure represents the mean over 50 executions.

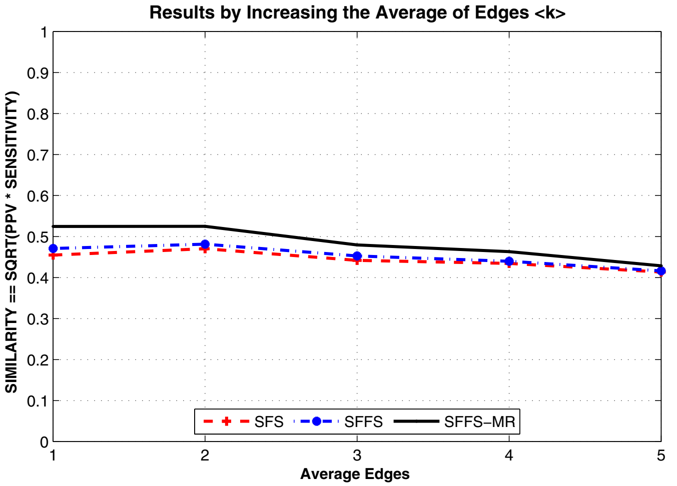


Fig. 3. Similarity measure, SQRT(PPV*Sensitivity), obtained by SFS, SFFS and SFFS-MR applied to infer network edges from different network complexities in terms of average degree $\langle k \rangle$. Similarity measure represents the mean over 50 executions.

(SFFS) and 51% (SFS) after only 20 observations and getting 75% against 57% (SFFS) and 54% (SFS) after 100 observations, even in the presence of some perturbations in the temporal signal, implied by the stochasticity in the application of transition functions.

The second experiment was performed in order to compare the robustness of the methods by increasing the complexity of the networks in terms of its average degree $\langle k \rangle$. The geometric mean between PPV and Sensitivity, presented in Fig. 3, was calculated by taking into account the average results for all variations of signal size. Fig. 3 shows that the three methods were very robust to increasing complexity of networks, presenting a soft decrease of similarity with the increase of average degree $\langle k \rangle$. In addition, the SFFS-MR showed slightly better results than the SFS and SFFS.

7 Conclusion

This work presents a floating search strategy for the inference of gene regulatory networks. Given the known limitations, our focus is to bring attention to the inclusion of prior knowledge on search methods, so that it occurs more efficiently. The proposed strategy is based on the assumption that some genes in biological organisms have an intrinsically multivariate prediction. The presented method exploits this property by the inclusion of multiple roots at the beginning of the search, which are defined by the best and worst single results of the SFS algorithm. In this context, the search space traversed by the SFFS-MR method is a little wider than SFS and SFFS, but does not worsen the asymptotical computational cost of the SFFS.

The experimental results show that the SFFS-MR provides a better inference accuracy (PPV) than SFS and SFFS, when considering small signal sizes with 15-20 time points and also with large ones, with 100 time points. In addition, the SFFS-MR was able to achieve 60% of accuracy on network recovery after only 20 observations from a state space of size 2^{20}, presenting very good results.

The SFFS-MR has also proved to be robust, as SFS and SFFS, when submitted to the increasing complexity of the networks in terms of its average degree $\langle k \rangle$. The robustness is an important property for the inference methods, even in the presence of some perturbations in the temporal signal, implied by the stochasticity in the application of transition functions. Besides, the SFFS-MR showed slightly better results than the SFS and SFFS.

A possible extension of the present work is to apply the SFFS-MR in order to evaluate large-scale networks, as well as to compare this method to other network inference methods based on feature selection. Also, we plan to apply this technique to infer GRNs from real data. Another very important improvement in search methods for GRNs inference would be the inclusion of topology information, such as the small-world (WS) [37] and scale-free (BA) [2] in order to guide the search process for the correct topology inference of these networks.

Acknowledgments

This work was supported by FAPESP, CNPq and CAPES.

References

1. Anastassiou, D.: Computational analysis of the synergy among multiple interacting genes. Molecular Systems Biology 3(83) (2007)
2. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science 286(5439), 509–512 (1999)
3. Barrera, J., Cesar Jr., R.M., Martins Jr., D.C., Vencio, R.Z.N., Merino, E.F., Yamamoto, M.M., Leonardi, F.G., Pereira, C.A.B., Portillo, H.A.: Methods of Microarray Data Analysis V. In: Constructing Probabilistic Genetic Networks of Plasmodium Falciparum, from Dynamical Expression Signals of the Intraerythrocytic Development Cycle, pp. 11–26. Springer, Heidelberg (2007)
4. Butte, A., Kohane, I.: Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In: Proceedings of the Pacific Symposium on Biocomputing, pp. 418–429 (2000)
5. D'haeseleer, P., Liang, S., Somogyi, R.: Genetic network inference: from co-expression clustering to reverse engineering. Bioinformatics 16(8), 707–726 (2000)
6. Dougherty, E.R.: Validation of inference procedures for gene regulatory networks. Current Genomics 8(6), 351–359 (2007)
7. Dougherty, E.R., Brun, M., Trent, J.M., Bittner, M.L.: Conditioning-Based Modeling of Contextual Genomic Regulation. IEEE/ACM TCBB 6(2), 310–320 (2009), `http://doi.ieeecomputersociety.org/10.1109/TCBB.2007.70247`
8. Dougherty, E.R., Kim, S., Chen, Y.: Coefficient of determination in nonlinear signal processing. Signal Processing 80, 2219–2235 (2000)
9. Dougherty, J., Tabus, I., Astola, J.: Inference of gene regulatory networks based on a universal minimum description length. EURASIP Journal on Bioinformatics and Systems Biology, 1–11 (2008)
10. Erdös, P., Rényi, A.: On random graphs. Publ. Math. Debrecen 6, 290–297 (1959)
11. Faith, J., Hayete, B., Thaden, J., Mogno, I., Wierzbowski, J., Cottarel, G., Kasif, S., Collins, J., Gardner, T.: Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles. PLoS Biology 5(1), 259–265 (2007)
12. Ghaffari, N., Ivanov, I., Qian, X., Dougherty, E.R.: A CoD-based reduction algorithm for designing stationary control policies on Boolean networks. Bioinformatics 26(12), 1556–1563 (2010) doi: 10.1093/bioinformatics/btq225, `http://bioinformatics.oxfordjournals.org/cgi/content/abstract/26/12/1556`
13. Hashimoto, R.F., Kim, S., Shmulevich, I., Zhang, W., Bittner, M.L., Dougherty, E.R.: Growing genetic regulatory networks from seed genes. Bioinformatics 20(8), 1241–1247 (2004)
14. Hecker, M., Lambeck, S., Toepfer, S., van Someren, E., Guthke, R.: Gene regulatory network inference: Data integration in dynamic models - A review. Biosystems 96(1), 86–103 (2009)
15. Hovatta, I., Kimppa, K., Lehmussola, A., Pasanen, T., Saarela, J., Saarikko, I., Saharinen, J., Tiikkainen, P., Toivanen, T., Tolvanen, M., et al.: DNA microarray data analysis. In: CSC, 2nd edn., Scientific Computing Ltd. (2005)
16. de Jong, H.: Modeling and simulation of genetic regulatory systems: A literature review. Journal of Computational Biology 9(1), 67–103 (2002)
17. Karlebach, G., Shamir, R.: Modelling and analysis of gene regulatory networks. Nat. Rev. Mol. Cell Biol. 9(10), 770–780 (2008)
18. Liang, S., Fuhrman, S., Somogyi, R.: Reveal: a general reverse engineering algorithm for inference of genetic network architectures. In: Proceedings of the Pacific Symposium on Biocomputing, pp. 18–29 (1998)

19. Lopes, F.M., Martins Jr., D.C., Cesar Jr., R.M.: Feature selection environment for genomic applications. BMC Bioinformatics 9(1), 451 (2008)
20. Lopes, F.M., Cesar Jr., R.M., Costa, L.d.F.: AGN simulation and validation model. In: Bazzan, A.L.C., Craven, M., Martins, N.F. (eds.) BSB 2008. LNCS (LNBI), vol. 5167, pp. 169–173. Springer, Heidelberg (2008)
21. Margolin, A., Basso, K.N., Wiggins, C., Stolovitzky, G., Favera, R., Califano, A.: ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics 7(suppl. 1), S7 (2006)
22. Martins Jr., D.C., Braga-Neto, U., Hashimoto, R.F., Dougherty, E.R., Bittner, M.L.: Intrinsically multivariate predictive genes. IEEE Journal of Selected Topics in Signal Processing 2(3), 424–439 (2008)
23. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE TPAMI 27(8), 1226–1238 (2005)
24. Pudil, P., Novovičová, J., Kittler, J.: Floating search methods in feature-selection. Pattern Recognition Letters 15(11), 1119–1125 (1994)
25. Steuer, R., Kurths, J., Daub, C., Weise, J., Selbig, J.: The mutual information: detecting and evaluating dependencies between variables. Bioinformatics 18(Suppl. 2), 231–240 (2002)
26. Rao, A., Hero III, A., States, D., Engel, J.: Using directed information to build biologically relevant influence networks. In: Proc. LSS Comput. Syst. Bioinform, pp. 145–156 (August 2007)
27. Ris, M., Martins Jr., D.C., Barrera, J.: U-curve: A branch-and-bound optimization algorithm for u-shaped cost functions on boolean lattices applied to the feature selection problem. Pattern Recognition 43(3), 557–568 (2010)
28. Schllit, T., Brazma, A.: Current approaches to gene regulatory network modelling. BMC Bioinformatics 8(suppl. 6), S9 (2007)
29. Shalon, D., Smith, S.J., Brown, P.O.: A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization. Genome Research 6(7), 639–645 (1996)
30. Shmulevich, I., Dougherty, E.R., Kim, S., Zhang, W.: Probabilistic boolean networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics 18(2), 261–274 (2002)
31. Somol, P., Pudil, P., Kittler, J.: Fast branch & bound algorithms for optimal feature selection. IEEE TPAMI 26(7), 900–912 (2004)
32. Somol, P., Pudil, P., Novovičová, J., Paclík, P.: Adaptive floating search methods in feature selection. Pattern Recognition Letters 20, 1157–1163 (1999)
33. Stuart, J.M., Segal, E., Koller, D., Kim, S.K.: A gene-coexpression network for global discovery of conserved genetic modules. Science 302(5643), 249–255 (2003)
34. Styczynski, M.P., Stephanopoulos, G.: Overview of computational methods for the inference of gene regulatory networks. Computers & Chemical Engineering 29(3), 519–534 (2005)
35. Velculescu, V.E., Zhang, L., Vogelstein, B., Kinzler, K.W.: Serial Analysis of Gene Expression. Science 270(5235), 484–487 (1995)
36. Wang, Z., Gerstein, M., Snyder, M.: RNA-Seq: a revolutionary tool for transcriptomics. Nat. Rev. Genet. 10(1), 57–63 (2009)
37. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world networks. Nature 393, 440–442 (1998)
38. Zhao, W., Serpedin, E., Dougherty, E.R.: Inferring connectivity of genetic regulatory networks using information-theoretic criteria. IEEE/ACM TCBB 5(2), 262–274 (2008)

Revisiting the Voronoi Description of Protein-Protein Interfaces: Algorithms

Frederic Cazals

INRIA Sopha-Antipolis, France
Frederic.Cazals@sophia.inria.fr
http://www-sop.inria.fr/abs/Frederic.Cazals

Abstract. Describing macro-molecular interfaces is key to improve our understanding of the specificity and of the stability of macro-molecular interactions, and also to predict complexes when little structural information is known. Ideally, an interface model should provide easy-to-compute geometric and topological parameters exhibiting a good correlation with important bio-physical quantities. It should also be parametric and amenable to comparisons. In this spirit, we recently developed an interface model based on Voronoi diagrams, which proved instrumental to refine state-of-the-art conclusions and provide new insights.

This paper formally presents this Voronoi interface model. First, we discuss its connexion to classical interface models based on distance cut-offs and solvent accessibility. Second, we develop the geometric and topological constructions underlying the Voronoi interface, and design efficient algorithms based on the Delaunay triangulation and the α-complex.

We conclude with perspectives. In particular, we expect the Voronoi interface model to be particularly well suited for the problem of comparing interfaces in the context of large-scale structural studies.

Keywords: Protein interfaces, Computational Geometry, Voronoi diagrams, Geometric patterns.

1 Introduction

1.1 On Classical Protein - Protein Interface Models

Modeling interfaces. Understanding the stability and the specificity of macro-molecular interactions is a key endeavour in computational structural biology. Such an endeavour requires on the one hand describing non-covalent interactions for the interfaces of complexes which have been solved experimentally, and on the other hand developing algorithms able to predict complexes when little or no structural information on the partners is known. On a per-complex basis, interface models allow one to investigate correlations between structural parameters and key bio-physical properties such as the conservation of residues, their polarity, the water dynamics at the interface, mutagenesis data, etc. For large scale experiments in the context of proteomics, the comparison of binding patches

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 419–430, 2010.

associated to interface models allow, in particular, the investigation of putative partners between orphan molecules.

Classical interface models. Classical interface models are twofold. May be the most widely used model is the so-called geometric footprint also called distance-based model, which consists of considering all pairs of atoms within a distance threshold d, typically in the range 5-8Å. But as illustrated on Fig. 1, considering all atoms of one partner which are within distance d from the second partner results in a bias towards convex regions [GLN04]. Another very popular interface model is that based on the Solvent Accessible Surface (SAS). Recall that the SAS is the boundary of balls with *expanded radii*, i.e. Van der Walls radii expanded by $r = 1.4\mathring{A}$ to account for a water probe. In this model, an interface atom is an atom contributing to the SAS of its own sub-unit, but losing part of this exposed surface in the complex. See Fig. 2. Interface atoms identified this way can further be classified as exposed or buried, depending on whether they retain accessibility in the complex, the former and latter making up the so-called rim and core of the interface. But as established in [CPBJ06] and explained in section 2.1, the SAS model actually omits privileged contacts.

Overall, a general drawback of these two models is that they do not provide a rich framework to compute pieces of information such as volume and surface areas, curvature information, dissection of the interface into patches. Instead, the computation of such quantities requires running dedicated algorithms.

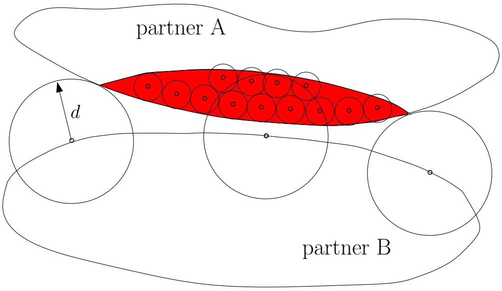


Fig. 1. Defining an interface based on a distance threshold creates a bias towards convex regions

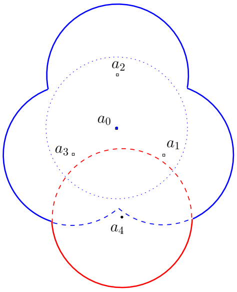


Fig. 2. Defining an interface from atoms losing solvent accessibility. The dashed regions are exposed in the red and blue sub-units, but get covered in the complex.

1.2 The Voronoi Interface

Intuitive presentation. Recall that the Euclidean Voronoi diagram of a collection of *sample* points is the partition of the ambient space into convex cells, such that all points in a cell have the sample associated to the cell as nearest neighbor. In bio-chemistry, since atoms' radii depend on their chemical type, one replaces the Euclidean distance by the so-called power distance[1]. Abusing terminology, we still call the resulting diagram a Voronoi diagram. In the sequel, we shall consider the Voronoi diagram of atoms with expanded radii. See Fig. 3 for a 2D illustration.

Neighbors in a Voronoi diagram are actually *privileged neighbors*. That is, given two neighboring spheres S_i and S_j, and for any point p found on the dual Voronoi face, one has $\pi(p, S_i) = \pi(p, S_j) < \pi(p, S_k), \forall k \neq i, j$. This property is the main incentive for using pairs of neighboring regions to report interface neighbors. However, two atoms may share a Voronoi face, yet their relative distance might be arbitrarily large. To get around this difficulty, let a *restricted ball* or *restriction* be the intersection of this ball with its Voronoi region, see e.g. the red ball on Fig. 3(a). Focusing on pairs of neighboring restrictions instead of pairs of neighboring Voronoi regions allows one to report pairs of privileged neighbors without resorting to a distance cutoff. We illustrate this construction to define our Voronoi interface model.

Consider a complex involving two partners A and B, and denote W the water molecules, if any. These types are also referred to as *colors*. Atoms of type $A/B/W$ are denoted $a_i/b_i/w_i$, respectively.

Let an *interface water molecule* be a water molecule such that its restriction has neighboring restrictions of type A and B. Water molecules which are not at the interface are called *bulk* water molecules. As illustrated on Fig. 3(a), our interface features pairs of restrictions of type $[A; B]$ or $[A; W]$ or $[B; W]$, with W standing for interface water molecules. Each pair actually conveys two pieces of information, namely the atoms associated to the restrictions, and the Voronoi facet, also called *tile*, dual of this edge. As illustrated on Fig. 3(b), an interface atom is an atom involved in at least one pair. Focusing on two types allows one to define three *bicolor* interfaces. That is, tiles of type AB (AW and BW) define the interface AB (AW and BW respectively). Tiles of type AB define direct contacts between the partners, while tiles of type AW and BW define contacts between these partners which are mediated by interface water molecules. The union of tiles AW and BW defines the AW–BW interface. The union of the AW–BW and AB interfaces defines the ABW interface, which separates the partners and gives a global overview of the interaction area, regardless of the role played by water molecules. See Fig. 3(c,d).

Interestingly, the interface ABW can be *shelled into concentric shells*—prosaically speaking the process is similar to peeling an onion from the outside to the inside. The process yields a integer called the *Voronoi Shelling Order* or VSO to tiles and atoms. This VSO qualifies the depth of an atom at the interface—from one for rim atoms to an integer in the range 7-10 for most complexes. See Fig. 3(c,d).

Connexions to classical interface parameters. We now discuss the finding made with our interface model, and note in passing that the corresponding

[1] Recall that the power distance of a point p to a sphere $S(c_i, r_i)$ is defined by $\pi(p, S_i) = \|c_i p\|^2 - r_i^2$.

software, Intervor [LC10], can be run and retrieved from http://cgal.inria.fr/abs/Intervor, together with plugins for VMD and Pymol.

We established in [CPBJ06] that our model identifies a superset of interface atoms losing solvent accessibility [BCR+04], which actually draws the attention to interactions between main chain atoms upon association. (Algorithms in section 2.1.) Interface tiles are naturally gathered into patches, which have been shown [CPBJ06] to be coherent with those obtained with classical clustering algorithms [CJ02]. (Algorithms in sections 2.2, 2.3 and 3.2.) Quantifying the planarity of interfaces and patches is important, e.g. to estimate (de-)solvation energies and also to identify putative binding regions for docking. While previous studies have used strategies based on plane fitting [JT96], the Voronoi interface comes with a notion of discrete (mean) curvature [CPBJ06], which allows to assess the curvature properties at any scale (from two tiles to the whole interface). (Algorithms in section 2.4.) Finally, the VSO provides a discrete interface depth parameterization which refines the dissection into a core and a rim [CCJ99, BCR+04]. In [BGNC09], this parameterization allowed us to sharpen the investigation of correlations between (i) the interface geometry, (ii) the location of polar residues [CJ02], (iii) the location of conserved residues [GC05], (iv) the dynamics of interfacial water [MRL07]. (Algorithms in section 3.3.)

We note in passing that another Voronoi interface definition has been proposed in [BER04]. This interface model uses the Voronoi diagram of the Van der Walls atoms (rather than the expanded radii), and closes small gaps at the interface using a growth process of the atoms which consists of expanding their squared radii by a value α. As a consequence, the atoms reported are not qualified with respect to solvent accessibility.

2 Bicolor Voronoi Interfaces

In this section, we formally define bicolor Voronoi interfaces. We use the terminology of bicolor AB interface —the interface between the two proteins A and B, although the presentation is identical for any bicolor interface. We assume that the reader is familiar with the α-complex of a union of balls [Ede92].

2.1 Bicolor Interface and Interface Neighbors

To account for privileged contacts in the Voronoi diagram of atomic balls, we seek pairwise intersections of restrictions of different colors. To balls whose restrictions intersect actually define an edge in the α-complex of the balls for $\alpha = 0$, whence the following:

Definition 1. *An AB* interface edge *is an edge of type AB in the α-complex of the balls B_i, with $\alpha = 0$; its endpoints are called* interface atoms. *The* interface neighbors *of a sphere S_i are the atoms of the second molecule sphere S_i is connected to through an interface edge.*

The AB interface *is defined as the collection Voronoi facets dual of the AB interface edges. A Voronoi edge bounding an interface Voronoi facet is called an* interface Voronoi edge.

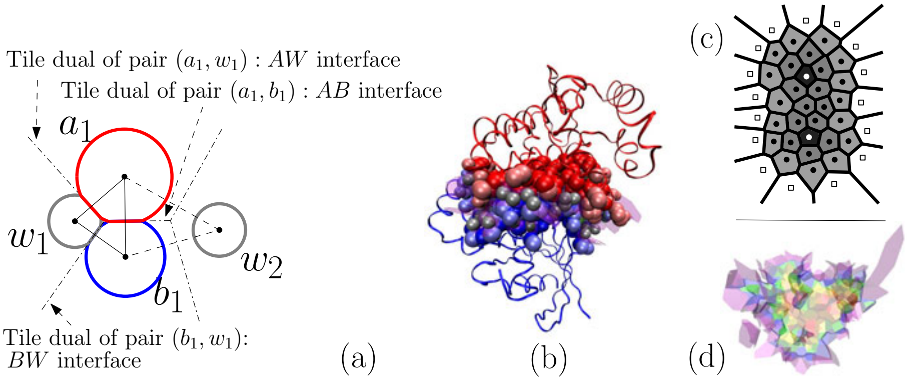

Tile dual of pair (a_1, w_1) : AW interface
Tile dual of pair (a_1, b_1) : AB interface
a_1
w_1
w_2
b_1
Tile dual of pair (b_1, w_1):
BW interface
(a)
(b)
(c)
(d)

Fig. 3. (a) A fictitious complex with two atoms (red and blue) and two water molecules (in grey). The Voronoi diagram consists of the dashed-dotted line-segments. The interface comprises three pairs namely $[a_1; b_1]$, $[a_1; w_1]$, and $[b_1; w_1]$; water w_2 is not at the interface. (b) Signal transduction complex (1tx4.pdb) : chains and interface atoms displayed with radii expanded by $1.4\AA$, with interface water molecules in grey. (c) Shelling a fictitious 2D interface into three shells (d) Shelling the ABW interface of complex 1txa into concentric shells: transparent view of the shells.

With respect to interface atoms defined with the BSA criterion, one can prove the following:

Observation 1. *Any atom S_i such that losing solvent accessibility during complex formation is an interface atom by Def. 1.*

However, the converse is false, as already mentioned in section 1.1, since an interface atom by Def. 1 may not lose solvent accessibility. A sufficient—but not necessary—condition for that is met when the atom is buried in its own subunit, and a 2D illustration is provided on Fig. 4. On that figure, the maintenance of the so-called *empty ball property* which characterizes Delaunay triangulations results in the creation an edge between the buried atom centered at a_0 and the red atom centered at a_4.

2.2 Topology of Bicolor Voronoi Interfaces

Consider a bicolor interface, say the AB interface. Since the interface is a subset of the Voronoi diagram, it is a cell complex. To further qualify its topology, we need to examine how Voronoi facets patch together. Since a facet is the dual of a bicolor edge, we examine the tetrahedra containing this edge. We define:

Definition 2. *The* type *of a tetrahedron featuring atoms of types A and B is denoted by a pair (i, j) where i, j respectively count the number of atoms of each type, and $i + j = 4$. Similarly, the type of a tetrahedron featuring an additional atomic type out of W, X is denoted by a triple (i, j, k), with k the number of atoms of the third type, and $i + j + k = 4$.*

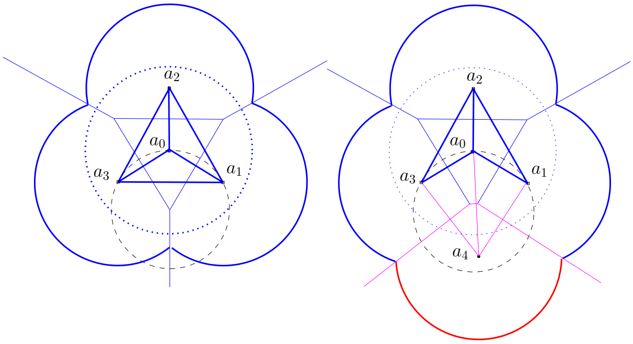


Fig. 4. In the Voronoi model, interface atoms can be buried (a) Atom centered at a_0 is buried in its subunit (b) Yet, it makes an interface edge with atom centered at a_4 in the complex

A case analysis of the types of tetrahedra yields:

Observation 2. *The bicolor interface has the following properties:*

- *A Voronoi edge bounding an AB Voronoi facet has one or two incident AB Voronoi facets. Such a Voronoi edge is on the interface boundary iff only one edge of its dual triangle is an interface edge of type AB.*
- *The neighborhood of every Voronoi vertex is either a topological disk, a half-topological disk, or two half-topological disks pinched together at the Voronoi vertex. This later case correspond to a tetrahedron of type* $(2, 2)$*, where two bicolor edges are interface edges, and these two edges span the four vertices of the tetrahedron.*

An interface can be connected through a pinched Voronoi vertex of a type $(2, 2)$ tetrahedron. For such a tetrahedron, out of the four bicolor edges, only two actually occur in the α-complex. Phrased differently, we have two independent bicolor pairs. See Figs. 5 and 6. Since the intersection of the corresponding Voronoi facets does not encode a joint property of these pairs, we define:

Definition 3. *Two Voronoi facets are called* edge-connected *if they share a Voronoi edge. An* edge-connected *component of the interface is a collection of edge-connected Voronoi facets. The closed curves bounding an edge connected component are called the* loops*.*

We now present the algorithms used to retrieve connected components and loops.

2.3 Computing Bicolor Interfaces and Their Boundaries

Retrieving connected components. Given an initial Voronoi facet f_0 dual of an edge e_0, the exploration of the corresponding connected component requires running a breadth-first-search (or depth-first-search) like algorithm anchored

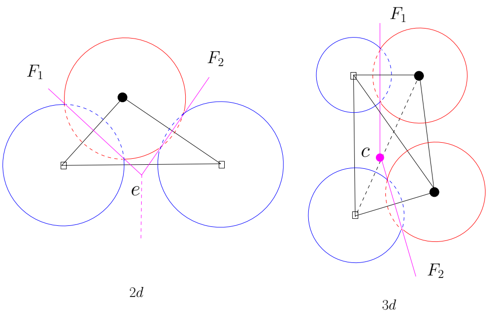


Fig. 5. (a) Top view of a Delaunay triangle in $3d$. Voronoi facets F_1 and F_2 meet along the Voronoi edge dual of the triangle, and are edge-connected (b)Side view of Delaunay tetrahedron contributing 2 bicolor interface edges. The dual Voronoi facets F_1 and F_2 meet at a pinched Voronoi vertex dual of a type $(2, 2)$ tetrahedron, and are not edge-connected.

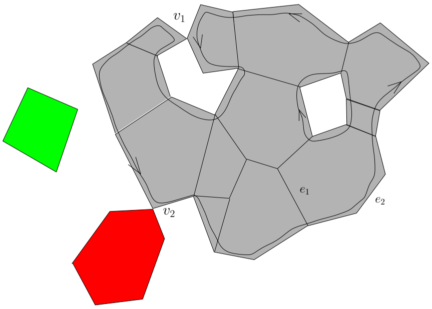


Fig. 6. An interface with three connected components and four boundary loops

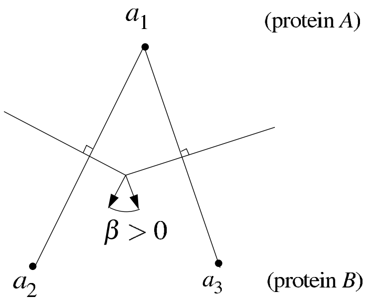


Fig. 7. Dihedral angle between Voronoi facets

at f_0. To run such an algorithm, the only information required is the list of Voronoi facets edge connected to a given facet f. But two edge connected Voronoi facets are dual of edges of the same triangle. Therefore, to report the facets edge connected to a facet f dual of an edge e, we just need to report the Delaunay triangles (i)incident to e (ii)featuring a second bicolor edge whose type is that of e. This is easily done by rotating around edge e in the Delaunay triangulation.

Upon completion of a connected component, we have a collection of Delaunay edges. We record them without any additional topological information as any such information is encoded in the Delaunay triangulation. As an example, we present an algorithm to retrieve the cycles bounding a connected component. This algorithm operates on the Delaunay triangulation, but takes constant time per boundary edge.

Retrieving the cycles bounding a connected component. Starting from an initial boundary Voronoi edge e, a given cycle (also called loop) can be computed by iteratively following the successor of e on the boundary of the connected component. Assume e is oriented and denote $s(e), t(e)$ the corresponding source and target Voronoi vertices, and let $T(v)$ be the tetrahedron associated with Voronoi vertex v. The successor of edge e is one of the four Voronoi edges dual of the facets of $T(t(e))$. Following observation 2, if the neighborhood of $t(e)$ is a half-topological disk, tetrahedron $T(t(e))$ has only two facets whose dual Voronoi edges are boundary edges. Since e is one of them, extending the loop requires retrieving and following the second one. On the other hand, if $t(e)$ is a pinched Voronoi vertex and if the four edges belong to the boundary of the connected component processed, there are three potential outgoing edges, but only one bounding the Voronoi facet which has e on its boundary. Again, extending the loop requires finding and following this edge.

Equipped with this extension operation, computing the loop of a given edge e_0 requires picking an arbitrary orientation for e_0, and following the boundary until the Voronoi vertex $T(s(e_0))$ is encountered again. To retrieve all the loops, we just have to iterate over the remaining boundary edges which are not already part of a loop.

The previous description actually eludes a difficulty, namely the way infinite tetrahedra are handled. (Recall these are the tetrahedra featuring a triangle of the convex hull of the Delaunay triangulation). When such a tetrahedron is encountered during the extension of a loop, we do not report its Voronoi center but the weighted circumcenter of the finite facet —that is the center of the smallest sphere orthogonal to the three spheres associated with the vertices of the facet.

2.4 Geometry of Connected Components

The most straightforward geometric statistics for an interface are its surface area and its boundary length. Apart from these, another interesting quantity is the interface curvature. Since a bicolor interface is a piecewise linear orientable surface, the natural way to characterize its *extrinsic* curvature consists of using the mean curvature. Notice that since we aim at studying the way a surface is embedded in $\mathbb{R}^3$, the extrinsic curvature is more suited than the intrinsic Gauss curvature, which is related to topological invariants and in particular boundary properties —cf the Gauss-Bonnet theorem. Recall that the mean curvature of a polyhedral surface is carried out along edges [San79], the amount of mean curvature attached to an edge being defined by $h(e) = \beta(e)l(e)$, with $l(e)$ the edge length and $\beta(e)$ the angle between the normals to the facets incident to e, the angle being counted positively (negatively) if e is convex (concave). We thus define $s_H = \sum_{e \in \mathrm{IVE}} h(e)$, with IVE the collection of interior Voronoi edges of the interface.

In a bio-chemical setting, one expects dihedral angles to alternate. Therefore, large values of s_H indicate that the interface facets bend in a coherent fashion at the interface scale. Notice that in case of interfaces with several connected components, the components must be oriented coherently for the sum to make sense. But since the angle between the Voronoi facets matches the angle between the corresponding Delaunay edges, implementing this constraint means initializing the orientation of all the components in the same way —e.g. from protein A to protein B. See Fig. 7.

2.5 On the Geometry of Interface Facets

As seen in section 1.2, bicolor edges selected from the 0-complex allow one to report interface neighbors without resorting to a distance cutoff. Phrased differently, long edges belong to the Delaunay triangulation but not the α-complex. This filtering mechanism avoids using explicit solvent molecules, a strategy often resorted to in applications deriving statistical potentials from the Delaunay triangulation. However, this filtering mechanism does not provide any control on the geometry of the interface Voronoi facets, and in particular, large facets are expected near the convex hull of the atoms centers. In other words, interface edges encode the topology of the interface but not its geometry.

To retrieve a relevant geometric information, we build upon the observation that boundary atoms do not play a major role from an energetic standpoint [BT98], so that one may discard selected boundary edges these atoms are involved in. One way to discard large Voronoi facets is the following. Recall than any simplex in the α-complex comes with a value $\overline{\mu}$ which gives the weight of its largest orthogonal ball [Ede92]. For an interface edge e, denoting w_e the weight of its smallest ball, one can therefore discard the edge if $\overline{\mu}/w_e \geq M$, with M a positive number. Since weights of balls are equal to their square radii, the condition amounts to saying that the radii of the balls are within a factor $\sqrt{M}$.

3 Tricolor Interfaces and Water Molecules

3.1 The *AW–BW* Interface

When considering an interface, an interesting question is the role played by structural water molecules [2]. As these water molecules are described from the crystal as protein atoms are, we also expand their radius by the quantity r_w. Notice again that this expansion aims at mimicking an implicit continuous layer of solvent molecules on the atoms found in the crystal —be they protein atoms of water molecules.

If one has three molecular types A, B, W, one can define three types of bicolor interfaces. But since we primarily care for the AB interface, contact of type AW and BW are of interest only when located near the AB interface, see Fig. 8. We therefore define:

Definition 4. *An* interface water molecule *is a ball of type W which is the vertex of at least one edge of type AW and at least one edge of type BW, both edges belonging to the α-complex of the balls B_i, with $\alpha = 0$. An AW (or BW) interface edge is an edge of type AW (or BW), with W an interface water*

[2] A water molecule is termed *structural* if it is as stable as the surrounding atoms. In a crystal structure, this can be assessed thanks to B-factors.

molecule. The AW (BW) interface is defined as the collection Voronoi facets dual of the AW (BW) interface edges.

A further refinement consists of aggregating Voronoi facets of type AW and BW:

Definition 5. *The AW–BW* interface *is the collection of Voronoi facets dual of edges of type AW or BW. A* connected component *of AW–BW interface is a collection of edge-connected Voronoi facets dual of interface edges of type AW or BW.*

To study the AW or the BW or the AW–BW interface, observe that edge connected Voronoi facets of types AW and BW are defined from bicolor, tricolor or quadricolor tetrahedra. Let us analyze the last two cases. In such a tetrahedron, we identify the labels A and B —since we do not report facets dual of such edges, so that the configurations found are those of bicolor tetrahedra. More precisely, a tetrahedron of type $AABW$ where AB edges are omitted yields the same topological configurations as a bicolor $(3, 1)$ tetrahedron for any bicolor interface. A tricolor tetrahedron of type $ABWW$ is similar to a bicolor $(2, 2)$ tetrahedron for any bicolor interface. Finally, a $ABWX$ tetrahedron is equivalent to a $(2, 1, 1)$ tetrahedron for any bicolor interface.

Therefore, the AW, the BW, and the AW–BW interfaces have the same topological properties as the AB interface i.e. are surfaces with possibly pinched vertices.

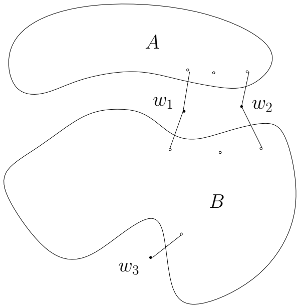


Fig. 8. Water molecules centered at w_1 and w_2 are interface water molecules; that centered at w_3 is not

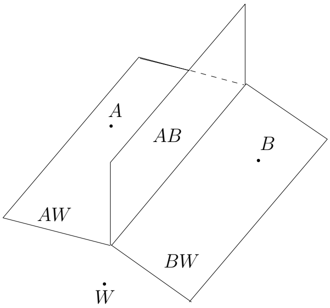


Fig. 9. The boundary of the union of the AB and AW–BW interfaces may not be a one-manifold

3.2 The *ABW* Interface

To assess the role of water molecules, and position relatively to one another the connected components of the AB and AW–BW interfaces, we define:

Definition 6. *The ABW* interface *is defined as the union of the AB and AW–BW interfaces.*

But the topology of the union is more involved than that of the singletons. First, non-manifold Voronoi edges may appear —if the three facets dual of the corresponding triangle are present in the union of the two interfaces. Second, the boundary of the union may not be a one-manifold, and we call it a *curve network* or net for short, see Fig. 9. To deal with these difficulties, it is actually sufficient to compute the AB and AW–BW interfaces separately, run a Union-Find algorithm to maintain the connected components of the edge-connected components, and another union-find algorithm to maintain the connected components of the boundary loops of the connected components of the union. Finding the Voronoi edges along which connexions occur can be done while computing the interfaces, while running m Union-Find operations on a n-element set takes $O(m\alpha(m,n)$ with $\alpha(m,n)$ the inverse of Ackerman's function [Tar83].

3.3 Shelling the *ABW* Interface

Considering the edge-connectivity of interface tiles, define the *depth* or the *Voronoi Shelling Order* of a tile as the number of tiles visited to reach it from the interface boundary—any tile which has a boundary edge is at depth one. This VSO provides an integer-valued parameterization of the ABW interface, which refines the binary core-rim model discussed in section 1.1.

4 Conclusion and Outlook

The interface model presented in this paper proved instrumental to refine our understanding of correlations between structural properties of protein interfaces, and important bio-physical quantities. However, the topic of modeling interfaces remains largely open for several reasons.

First, as evidenced by the scoring round of the community-wise docking experiment CAPRI, the design of scoring functions is an active area of research, and structural parameters defined from interface models should prove instrumental in this context.

Second, the question of precisely aligning and comparing interfaces has barely been touched upon. The Voronoi interface model proved instrumental for the description of bio-chemical properties, and we believe that the precise topological and geometric information it encodes should ease the comparison of interfaces in exhaustive structural classification studies.

Finally, while modeling single molecules and complexes is done routinely using methods from potential theory (molecular dynamics simulations, normal modes), we are not aware of any significant work for the problem of modeling dynamic interfaces so as to possibly incorporate entropy-related terms into scoring functions.

References

[BCR$^+$04] Bahadur, R., Chakrabarti, P., Rodier, F., Janin, J.: A dissection of specific and non-specific protein–protein interfaces. JMB 336(4), 943–955 (2004)

[BER04] Ban, Y.-E.A., Edelsbrunner, H., Rudolph, J.: Interface surfaces for protein-protein complexes. In: RECOMB, San Diego, pp. 205–212 (2004)

[BGNC09] Bouvier, B., Grünberg, R., Nilges, M., Cazals, F.: Shelling the voronoi interface of protein-protein complexes reveals patterns of residue conservation, dynamics and composition. Proteins 76(3), 677–692 (2009)

[BT98] Bogan, A.A., Thorn, K.S.: Anatomy of hot spots in protein interfaces. J. Mol. Biol. 280 (1998)

[CCJ99] Lo Conte, L., Chothia, C., Janin, J.: The atomic structure of protein-protein recognition sites. JMB 285(5), 2177–2198 (1999)

[CJ02] Chakrabarti, P., Janin, J.: Dissecting protein-protein recognition sites. Proteins 47(3), 334–343 (2002)

[CPBJ06] Cazals, F., Proust, F., Bahadur, R., Janin, J.: Revisiting the voronoi description of protein-protein interfaces. Protein Science 15(9), 2082–2092 (2006)

[Ede92] Edelsbrunner, H.: Weighted alpha shapes. Technical Report UIUCDCS-R-92-1760, Dept. Comput. Sci., Univ. Illinois, Urbana, IL (1992)

[GC05] Guharoy, M., Chakrabarti, P.: Conservation and relative importance of residues across protein-protein interfaces. PNAS 102(43), 15447–15452 (2005)

[GLN04] Grünberg, R., Leckner, J., Nilges, M.: Complementarity of structure ensembles in protein-protein binding. Structure 12(12), 2125–2136 (2004)

[JT96] Jones, S., Thornton, J.M.: Principles of protein-protein interactions. PNAS 93(1), 13 (1996)

[LC10] Loriot, S., Cazals, F.: Modeling macro-molecular interfaces with intervor. Bioinformatics 26(7), 964–965 (2010)

[MRL07] Mihalek, I., Reš, I., Lichtarge, O.: On itinerant water molecules and detectability of protein–protein interfaces through comparative analysis of homologues. JMB 369(2), 584–595 (2007)

[San79] Santaló, L.: Integral Probability and Geometric Probability. Encyclopedia of Mathematics and its Applications, vol. 1. Addison-Wesley, Reading (1979)

[Tar83] Tarjan, R.E.: Data Structures and Network Algorithms. CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 44. Society for Industrial and Applied Mathematics, Philadelphia (1983)

MC^4: A Tempering Algorithm for Large-Sample Network Inference

Daniel James Barker[1,2], Steven M. Hill[1,3], and Sach Mukherjee[3,1]

[1] Centre for Complexity Science, University of Warwick, Coventry, U.K. CV4 7AL
[2] Department of Physics, University of Warwick, Coventry, U.K. CV4 7AL
[3] Department of Statistics, University of Warwick, Coventry, U.K. CV4 7AL

Abstract. Bayesian networks and their variants are widely used for modelling gene regulatory and protein signalling networks. In many settings, it is the underlying network structure itself that is the object of inference. Within a Bayesian framework inferences regarding network structure are made via a posterior probability distribution over graphs. However, in practical problems, the space of graphs is usually too large to permit exact inference, motivating the use of approximate approaches. An MCMC-based algorithm known as MC^3 is widely used for network inference in this setting. We argue that recent trends towards larger sample size datasets, while otherwise advantageous, can, for reasons related to concentration of posterior mass, render inference by MC^3 *harder*. We therefore exploit an approach known as parallel tempering to put forward an algorithm for network inference which we call MC^4. We show empirical results on both synthetic and proteomic data which highlight the ability of MC^4 to converge faster and thereby yield demonstrably accurate results, even in challenging settings where MC^3 fails.

1 Introduction

Modern biochemical technologies are allowing access to ever increasing amounts of data pertaining to cellular processes. As a result, there has been a move away from studying molecular components in isolation towards pathway- and network-oriented approaches. This in turn has driven much work on network models in bioinformatics, machine learning and computational statistics. Probabilistic graphical models [6,5] have emerged as a key approach. These are stochastic models in which a graph is used to describe relationships between random variables and thereby facilitate representation and inference. Directed graphical models called Bayesian networks (BNs) are widely used in the modelling of gene regulatory and protein signalling networks [4,1,16,19].

A BN consists of a directed acyclic graph (DAG) G which describes conditional independence relationships between variables, and associated parameters. In many bioinformatics settings, it is of interest to make inferences about the DAG itself, a task known as structure learning or network inference.

Within a Bayesian framework, under certain assumptions, it is possible to analytically integrate out parameters to obtain a score which is *proportional*

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 431–442, 2010.

to the posterior probability of a given graph G. However, since the number of possible DAGs grows super-exponentially [15] it rapidly becomes infeasible to characterise the posterior distribution by simply enumerating all possible DAGs. This has motivated the use of approximate inference methods for network inference. Markov chain Monte Carlo (MCMC) methods [7,14] in particular are often used in this setting [4,10].

A widely-used approach is to follow [8] in using a random-walk Metropolis type algorithm in which moves are made in the state space of DAGs via single-edge changes (for details see Section 2 below). This scheme, known as MC^3 (for "Markov Chain Monte Carlo Model Composition"), is asymptotically guaranteed to converge to the desired posterior distribution, but can be slow to do so, and for large or otherwise challenging distributions can fail entirely (we show examples below).

In recent years, there has been a drive towards larger sample-size datasets for network inference. As the cost of array-based assays continues to fall, studies have become wider in scope, covering a greater number of samples. At the same time, single-cell, FACS-based platforms have also become more widely available. Clearly, this trend towards larger datasets is broadly favourable. Yet at the same time, it can render MCMC-based network inference *more* challenging. This is because as the sample size increases, the posterior mass becomes more concentrated around fewer graphs (eventually, by consistency of Bayesian model selection, around members of the correct, data-generating equivalence class). In this setting, the MC^3 scheme can have difficulty discovering these high-scoring graphs, or moving between them. Figure 1 shows an empirical example of this phenomenon. As we increase the sample-size N for a simple, four-node toy-model, the posterior becomes progressively more concentrated on a few graphs. (Note, locality of graphs along the axis in Figure 1 does not represent location in graph space).

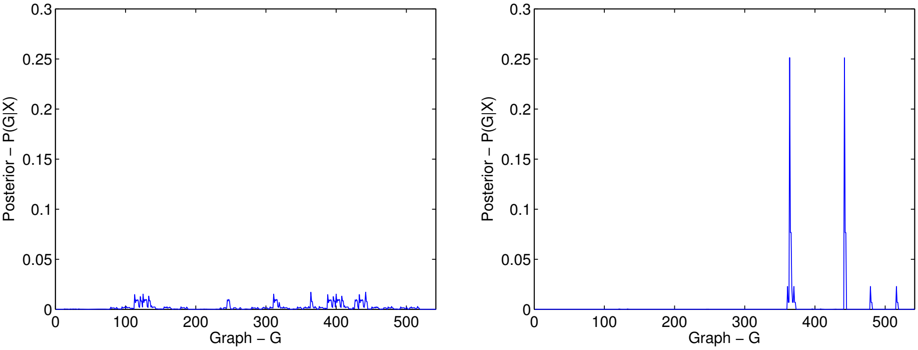


Fig. 1. The posterior distribution of graphs $P(G|\mathbf{X})$ for two lengths of data set N. As N increases the distribution becomes more peaked on a few highly probable graphs. The distributions shown here are over the space of 4-node DAGs; a space small enough to permit enumeration of the true distribution.

This 'peakiness' can be quantified by the information entropy

$$H[P(G|\mathbf{X})] = -\sum_{G \in \mathcal{G}} P(G|\mathbf{X}) \log P(G|\mathbf{X}) \tag{1}$$

where, $P(G|\mathbf{X})$ denotes the posterior probability of a graph G given data $\mathbf{X}$ and $\mathcal{G}$ is the whole space of DAGs. Using Bayes' theorem we can write the posterior as proportional to the product of a marginal likelihood, $P(\mathbf{X}|G)$ and a prior distribution over graphs, $P(G)$;

$$P(G|\mathbf{X}) \propto P(\mathbf{X}|G)P(G). \tag{2}$$

The marginal likelihood is obtained by integrating out model parameters Θ. The likelihood $P(\mathbf{X}|G, \Theta)$ factorises into a product of local terms in which each variable X_i depends only on its parents in graph G, $\pi_G(i)$, and parameters θ_i. That is, $P(\mathbf{X}|G, \Theta) = \prod_i P(X_i|\pi_G(i), \theta_i)$.

Entropy H and shape of the posterior are affected by both marginal likelihood and graph prior. The main focus of the present paper are low-entropy regimes that are characteristic of large-sample problems. We therefore focus only on the effect of increasing sample size and choose a uniform graph prior, $P(G) = |\mathcal{G}|^{-1}$.

H is maximal for a uniform distribution so its maximal possible value is $H_{\max} = \log |\mathcal{G}|$. Consider the information entropy of the distributions on 4-node DAGs shown in Figure 1; as N increases we move away from the uniform distribution which is reflected by H decreasing from $H_{\max} \simeq 6.3$ to a lower value of $\simeq 1.5$ (see Figure 2).

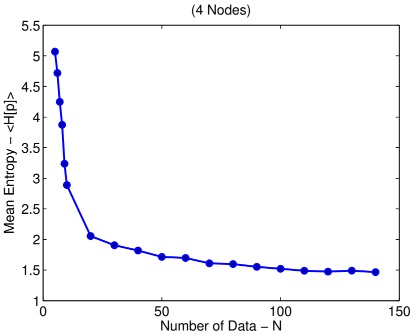


Fig. 2. The information entropy $H[p]$ (averaged over 100 subsamplings of a data set) as a function of length of the data set N. We can see that as we increase N the information entropy H decreases indicating that we are moving further away from the uniform distribution.

Low-entropy regimes of this kind, which are important for the larger datasets that are now becoming available, present special challenges for MCMC. One approach, popular in both statistical physics and Bayesian statistics, is to permit either longer-range or, via so-called tempering algorithms, "higher temperature" moves around the state space. Here, we show how a form of tempering known as parallel tempering can be used to accelerate convergence of MCMC-based network inference. The approach has a minimum of user-set parameters and is amenable to efficient, parallel implementation, giving effective run-times identical to MC^3. We show comparative results on both synthetic data and data from high-throughput proteomics. Since tempering approaches are often referred to as "Metropolis-coupled", we call our approach "MC^4" for "Model Composition by Metropolis-Coupled Markov Chain Monte Carlo".

The remainder of the paper is organised as follows. We first introduce notation and briefly review MCMC-based network inference. We then introduce the parallel tempering scheme used and illustrate how it can help inference in the relatively low entropy regimes of interest here. We then show empirical results comparing relative performance on both synthetic and proteomic data. We conclude with a discussion of the work presented and ideas for future work.

2 Methods

2.1 Monte Carlo Schemes

The Monte Carlo schemes used here can be thought of as constructing a Markov chain whose state space is the space of DAGs $\mathcal{G}$ and whose (unique) invariant distribution is the posterior distribution $P(G|\mathbf{X})$ of interest. For a more detailed technical exposition of these ideas we refer the interested reader to [14] and references therein.

Given $t_{\max}$ samples our estimate of the probability of a graph G is given by

$$\hat{P}(G|\mathbf{X}) = \frac{1}{t_{\max}} \sum_{t=1}^{t_{\max}} I(g^{(t)} = G) \tag{3}$$

where $g^{(t)}$ is the t^{th} sampled graph and $I(\cdot)$ is the indicator function which equals 1 if its argument is true and 0 otherwise. For MCMC schemes which satisfy certain mild conditions we also have, by standard results:

$$\lim_{t_{\max}\to\infty} \frac{1}{t_{\max}} \sum_{t=1}^{t_{\max}} I(g^{(t)} = G) = P(G|\mathbf{X}). \tag{4}$$

The Markov Chain performs its walk by proposing a new graph from the state space according to some 'proposal distribution' and then subsequently accepting or rejecting the proposed graph according to an 'acceptance probability', thereby ensuring the stationary distribution is the one desired.

MC^3. Here, the proposal distribution Q involves picking so-called 'neighbours' with uniform probability. The neighbourhood $\eta(G)$ of a graph G is defined to be all graphs G' which can be obtained from G by either removing, adding or

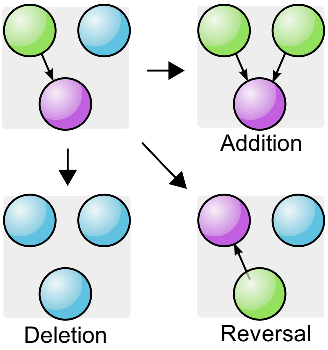


Fig. 3. The neighbourhood $\eta(G)$ of any directed acyclic graph G is defined as all acyclic graphs which are reachable from the current graph with one of the three basic edge operations; addition, deletion and reversal

flipping a single edge, whilst maintaining acyclicity (see Figure 3). The proposal distribution is then

$$Q(G \rightarrow G') = \begin{cases} \frac{1}{|\eta(G)|} & \text{if } G' \in \eta(G) \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

This proposal distribution has only short range support. It is worth noting that this proposal distibution is not symmetric since the sizes of the neighbourhoods, $|\eta|$, are (possibly) different for G and G'. However this is accounted for by a corresponding factor in the acceptance probability below which ensures that detailed balance is satisfied. This in turn is a sufficient condition to guarantee convergence to the correct posterior distribution in the limit. Specifically, the acceptance probability has the form $A = \min\{1, \alpha\}$ where

$$\alpha = \frac{P(\mathbf{X}|G')P(G')}{P(\mathbf{X}|G)P(G)} \frac{Q(G' \rightarrow G)}{Q(G \rightarrow G')} = \frac{P(\mathbf{X}|G')|\eta(G)|}{P(\mathbf{X}|G)|\eta(G')|}. \tag{6}$$

Here, the prior terms $P(G)$ and $P(G')$ cancel since we are using a uniform prior.

MC^4. In light of the concerns with MC^3 highlighted in the Introduction above, a natural idea is to consider "higher temperature" moves; a strategy which is widely used in statistical physics. Here, we use an approach known as Parallel Tempering (PT) to this end. PT is an MCMC-approach which aims to help the Markov chain escape local maxima, thus aiding mixing [7,12]. PT is a natural progression from work done by Marinari and Parisi [9] and Geyer and Thompson [3] on so-called Simulated Tempering (ST). In this statistical application there is no equivalent to the physical temperature, but we can introduce an analogue by writing

$$\alpha = \left(\frac{P(\mathbf{X}|G')P(G')}{P(\mathbf{X}|G)P(G)} \frac{|\eta(G)|}{|\eta(G')|} \right)^{\beta} \tag{7}$$

Here $\beta = T^{-1}$ is the inverse temperature. Clearly as $\beta \to 0$ (infinite temperature) $\alpha = 1$ and so we have the uniform distribution one would expect. Similarly as $\beta \to \infty$ (zero temperature) $\alpha = \infty$ if G' is more likely or $\alpha = 0$ if G is less likely and we recover steepest ascent.

In PT we run a collection of Markov Chains at different temperatures, occasionally swapping the graphs between them. To be more concrete, we have m chains each with an associated temperature T_i (β_i). The temperatures must obey $T_1 = 1$ ($\beta_1 = 1$) and $T_i > T_j$ ($\beta_i < \beta_j$) for $1 \leq j < i \leq m$. The algorithm for updating the chains is

(1) With probability $(1 - p_{\text{swap}})$ conduct a parallel step;
   (a) Update each graph G_i for each chain i using the MH scheme at temperature β_i.
(2) else conduct an exchange step;
   (a) Randomly choose a neighbouring pair of chains (i,j). Propose swapping their graphs G_i with G_j.
   (b) Accept the swap with probability $R = \min\{1, \rho\}$

$$\rho = \frac{(P(\mathbf{X}|G_j)P(G_j))^{\beta_i}\ (P(\mathbf{X}|G_i)P(G_i))^{\beta_j}}{(P(\mathbf{X}|G_i)P(G_i))^{\beta_i}\ (P(\mathbf{X}|G_j)P(G_j))^{\beta_j}}. \tag{8}$$

This scheme satisfies detailed balance in the extended state space thus convergence for each chain is guaranteed to the correct posterior distribution for each temperature [3,7].

The performace of this scheme depends upon the choice of temperatures and (somewhat more weakly) upon the exchange probability p_{swap}. A guide for choosing suitable temperatures is given in [7] as

$$(\beta_i - \beta_{i+1})\, |\Delta \log P| \approx - \log p_a \tag{9}$$

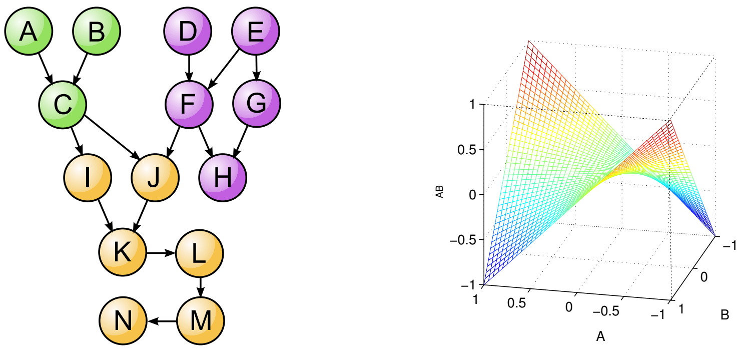
where $|\Delta \log P|$ is the typical difference in the log-posterior and p_a is the desired lower bound for the swapping acceptance probability.

Thanks to modern parallel computing facilities the updating of chains in step (1)(a) can be carried out simultaneously meaning this scheme can be run at effectively the same speed as the traditional MC^3 scheme, so long as the number of available processors is $\geq m$. If this condition is not satisfied we must wait for the chains to update before preceeding.

2.2 Simulation Set-Up

We run the two schemes on data simulated from the known network shown in Figure 4(a). Since we know the underlying network structure, we are able to assess and compare performance of the schemes.

Continuous data is generated by sampling the root nodes from a zero-mean Gaussian. Child nodes are also Gaussian distributed, but with mean dependent



(a) Data generating network.

(b) Continuous !XOR function.

Fig. 4. Simulation set-up. (a) Data generating network. (b) The cross terms in the known model (10) approximates the !XOR Boolean function and make it tough to infer the underlying structure (a).

on their parents in the graph. Specifically, when a node has a single parent, the mean is simply its parent's value. If there are two parents the mean of child is taken to be a non-linear combination of the parents. Thus the cumulative probabilities are defined as

$$P(A \leq x) = P(B \leq x) = \Phi\left(\frac{x}{\sigma}\right)$$

$$P(C \leq x | A, B) = \Phi\left(\frac{x - (A + B + \gamma AB)}{\sigma}\right) \tag{10}$$

for child node C with parents A and B, where $\Phi(x) = \frac{1}{2}[1 + \text{erf}(\frac{x}{\sqrt{2}})]$ is the cumulative distribution function of a standard Gaussian. The non-linear cross term γAB in the mean is chosen in the hope of separating the peaks of the distribution. If A is high and B is low (or vice versa) then C is low, if however A and B are both high (or low) then crucially C is high. This structure (illustrated in Figure 4) approximates the !XOR Boolean function, rendering difficult inference of the relationship between parents A and B and child C.

In order to investigate the effects of sample size N on the two schemes, we consider data sets with $N = 500$ and $N = 5,000$. We consider performance over ten MCMC runs of $t_{max} = 50,000$ iterations each; this gives good indications of convergence using standard diagnostics [2].

This paper concerns MCMC methods and the approaches we discuss apply to essentially any prior specification which yields a closed-form marginal likelihood or one which can be efficiently approximated. In all the experiments here we use a Gaussian model. Specifically, we take $X_i \sim \mathcal{N}(\mathbf{B}_i\beta_i, \sigma^2 I)$ where B_i is a local design matrix (including products over parents) and β_i corresponding regression coefficients. We use conjugate parameter priors [17,13] $\beta_i \sim \mathcal{N}(\mathbf{0}, N\sigma^2(\mathbf{B}_i^\mathsf{T}\mathbf{B}_i)^{-1})$ and $\sigma^2 \propto 1/\sigma^2$ to obtain the following closed form marginal likelihood,

$$p(\mathbf{X}|G) \propto \prod_i (1+N)^{-(2^{|\pi_G(i)|}-1)/2} \Big({X_i}^\mathsf{T} X_i \\ - \frac{N}{N+1} {X_i}^\mathsf{T} \mathbf{B}_i \left({\mathbf{B}_i}^\mathsf{T} \mathbf{B}_i \right)^{-1} {\mathbf{B}_i}^\mathsf{T} X_i \Big)^{-\frac{N}{2}} . \tag{11}$$

2.3 Measuring Convergence

We are interested in the marginal posterior probabilities for individual edges. We collect these probabilities into an "edge probability matrix" **E**, specifically:

$$\mathbf{E}_{ij} = \sum_{G \in \mathcal{G}} P(G|\mathbf{X}) I(e = (i,j) \in G). \tag{12}$$

We will also index entries in **E** by edge, e.g. $\mathbf{E}(e)$. Similarly to equation (3), we estimate the the edge probability of edge e by counting how many times it appears in the sampled graphs $g^{(t)}$,

$$\mathbf{E}_{ij}^{\mathrm{MC}} = \frac{1}{t_{\max}} \sum_{t=1}^{t_{\max}} I(e = (i,j) \in g^{(t)}). \tag{13}$$

If exact edge probabilities are available (as with our proteomic data study below), they can be used in tandem with our estimated edge probabilities to assess convergence. We use two measures of how well our Markov chains are converging; the correlation coefficient ρ between the exact and estimated edge probabilities and the normalised sum of absolute differences

$$S = \frac{1}{\nu} \sum_e |\mathbf{E}^{\mathrm{MC}}(e) - \mathbf{E}(e)| \tag{14}$$

where the sum runs over all possible edges and ν is the number of possible edges.

3 Results

3.1 Simulation Results

We assessed performance by thresholding posterior edge probabilities to obtain a set of edges and comparing this set to the true data generating graph edge set. We constructed receiver operating characteristic (ROC) curves for the MC^3 and MC^4 schemes with sample sizes $N = 500$ and $N = 5000$. The ROC curves show the number of false positives (edges obtained after thresholding that are not in the true graph) encountered for a given number of true positives (edges obtained after thresholding that are in the true graph). The curve is parameterised by the

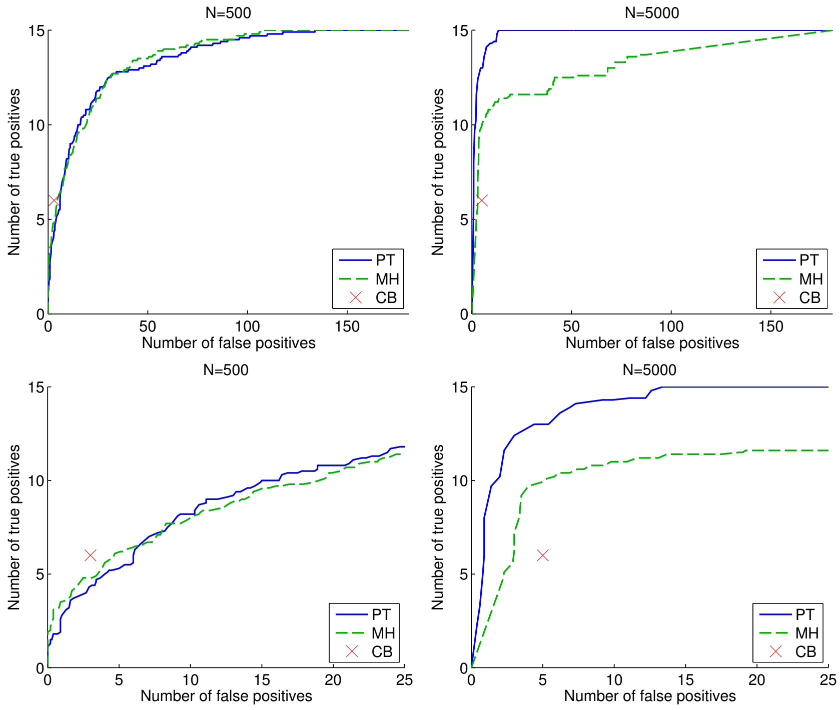


Fig. 5. Receiver operating characteristic (ROC) curves for simulated data. Number of true positive against number of false positive edge calls, produced by thresholding posterior edge probabilities at varying levels and comparing with the known, true data-generating graph. Results shown for MC^4 (blue solid line), MC^3 (green dashed line) and a recent deterministic constraint-based (CB) method for learning DAGs [18] (red cross), for sample sizes $N = 500$ (left) and $N = 5000$ (right). Lower panels show detail of corresponding upper panels. (MCMC results shown are averages over ten iterations).

threshold level. Figure 5 shows average ROC curves, produced by averaging ROC curves obtained from ten MCMC runs.

The area under the ROC curve (AUC) gives a simple measure of performance. Higher AUC values indicate superior accuracy. At the smaller sample size of $N = 500$ we see that MC^4 performs comparably with MC^3. The mean AUC value ($\pm$ standard deviation) for MC^4 is 0.90 ± 0.02 compared with 0.91 ± 0.02 for MC^3. However, at the larger sample size of $N = 5000$, we see a substantial improvement of performance with the MC^4 scheme compared with MC^3. The mean AUC value for MC^4 here is 0.99 ± 0.003, whereas the mean AUC value for MC^3 has decreased to 0.88 ± 0.13. This clearly illustrates that higher sample sizes can have a deleterious effect on the MC^3 scheme, whereas the MC^4 scheme improves dramatically in performance. At smaller sample sizes, with higher entropy posterior distributions, MC^3 performs well with MC^4 not providing any real gains.

We also compared our MCMC methods to a recent deterministic, constraint-based (CB) algorithm for learning DAGs [18] (default settings, significance level set to 0.05). For the $N = 500$ case, it performs comparably with MC^3 and MC^4 in terms of numbers of true and false positives. However, for the large-sample, $N = 5000$ case, we find that for the same number of false positives, MC^4 returns more than twice as many true positive edges as the CB algorithm.

3.2 Real Data Results

To investigate the performance of MC^4 on challenging experimental data, we make use of proteomic data from an ongoing study of cell signalling (unpublished data). Here the models are dynamic Bayesian networks (DBNs) [11] with 20 nodes (and thus 400 possible edges). The true underlying networks are not known in this case. However, by taking advantage of a certain factorisation of the graph space we can, in this case, calculate the edge probabilities *exactly*. The availability of exact results enables us to properly assess the performance of the MCMC schemes for this problem. We note that in practice, in this particular setting, one should use the exact calculation rather than an MCMC estimate. However, this design provides an ideal opportunity to test the MCMC methods on a large state-space problem based on real data but with gold-standard results available for comparison.

When applied to the real proteomic data we can see that parallel tempering provides a clear advantage over Metropolis-Hastings (see Figure 6). Both measures used to assess convergence are favourable for MC^4; the correlation between the exact edge probabilities and those estimated is closer to 1 for MC^4 than MC^3 and the per edge error as quantified by S is lower for MC^4.

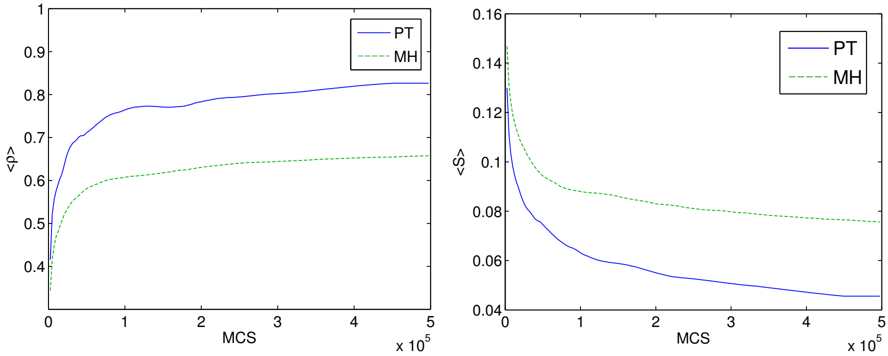


Fig. 6. Average correlation $\langle \rho \rangle$ between the exact edge probabilities and estimated edge probabilities for MC^4 and MC^3 (left) and the average per edge deviation error $\langle S \rangle$ (right). We can see from both measures that, in this real problem, MC^4 is outperforming MC^3 in terms of convergence. The parallel tempering used here had 5 temperatures evenly spaced between 1.0 and 2.0 with an exchange probability $p_{\mathrm{swap}} = 0.1$.

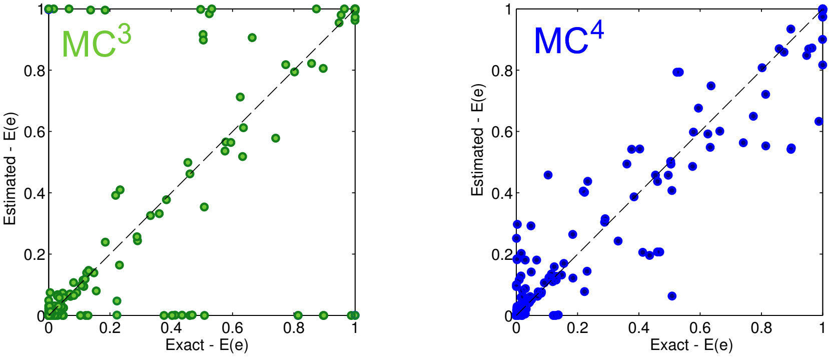


Fig. 7. Scatterplots of the exact edge probabilities versus MCMC estimated edge probabilities after $T = 500,000$ iterations for the real proteomic data for both MC^3 and MC^4 schemes. Notice that in the MC^3 case many of the edges lie in the off-diagonal corners $(0, 1)$ and $(1, 0)$, representing the most dramatic failures of the network inference. Use of the MC^4 scheme has remedied this with the offending edges in MC^3 being pulled significantly closer to the line $x = y$.

The scatter plots shown in Figure 7 serve to further elucidate this point. If the edges had been inferred perfectly they would lie on the line $x = y$ (representing $S = 0$), the farther points lie away from this line the worse our estimate of their value is. This means that points lying in the off-diagonal corners, as seen with MC^3, represent dramatic failures of inference. We observe that MC^4 has remedied this defect.

4 Conclusions

We have argued that MCMC-based network inference from larger datasets poses special problems for the widely-used MC^3 algorithm. As experimental designs become more ambitious in scope, better MCMC approaches will become ever more crucial for robust network inference. Motivated by these concerns, and by the increasing importance of inference in larger sample size settings, we proposed a tempering-based approach to network inference which we called MC^4.

We showed that MC^4 was able to outperform MC^3 in experiments on both simulated and real data, in some cases offering dramatic gains. These results support the idea that chains at higher temperatures can help inference in the regimes of interest by moving more freely in the regions of low scoring graphs. By coupling higher temperature chains to the desired target chain at $T = 1$, using exchange moves, we allowed it to move between the peaks while sampling them with the correct frequencies. In conclusion, the MC^4 algorithm put forward here is simple, requires little user-input and is demonstrably effective for network inference.

Acknowledgements. The authors are grateful to three referees for constructive comments which improved the paper; to Robert Goudie, Mario Nicodemi and Nicholas Podd for discussions and to EPSRC for support.

References

1. Friedman, N.: Inferring cellular networks using probabilistic graphical models. Science 303(5659), 799–805 (2004)
2. Gelman, A., Rubin, D.B.: Inference from iterative simulation using multiple sequences. Stat. Sci. 7, 457–472 (1992)
3. Geyer, C., Thompson, E.: Annealing Markov Chain Monte Carlo with Applications to Ancestral Inference. Journal of the American Statistical Association 90(431), 909–920 (1995)
4. Husmeier, D.: Reverse engineering of genetic networks with Bayesian networks. Biochemical Society Transactions 31(6), 1516–1518 (2003)
5. Jordan, M.: Graphical Models. Stat. Sci. 19, 140–155 (2004)
6. Lauritzen, S.: Graphical Models. O.U.P., Oxford (1996)
7. Liu, J.: Monte Carlo Strategies in Scientific Computing. Series in Statistics. Springer, New York (2008)
8. Madigan, D., York, J., Allard, D.: Bayesian Graphical Models for Discrete Data. International Statistical Review/Revue Internationale de Statistique 63(2), 215–232 (1995)
9. Marinari, E., Parisi, G.: Simulated Tempering: a New Monte Carlo Scheme. Europhys. Lett. 19(6), 451–458 (1992)
10. Mukherjee, S., Speed, T.: Network Inference Using Informative Priors. PNAS 105(38), 14313–14318 (2008)
11. Murphy, K.: Dynamic Bayesian Networks: Representation, Inference and Learning. Ph.D. thesis, Computer Science Division, Berkeley CA (2002)
12. Newman, M., Barkema, G.: Monte Carlo Methods in Statistical Physics. O.U.P., Oxford (1999)
13. Nott, D.J., Green, P.J.: Bayesian variable selection and the swendsen-wang algorithm. J. Comput. Graph. Stat. 13, 141–157 (2004)
14. Robert, C., Casella, G.: Monte Carlo Statistical Methods. Springer, New York (2004)
15. Robinson, R.: Counting Labeled Acyclic Digraphs. In: New Directions in the Theory of Graphs, pp. 239–273. Academic Press, London (1973)
16. Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D.A., Nolan, G.P.: Causal protein-signaling networks derived from multiparameter single-cell data. Science 308(5721), 523–529 (2005)
17. Smith, M., Kohn, R.: Nonparametric regression using Bayesian variable selection. J. Econometrics 75, 317–343 (1996)
18. Xie, X., Geng, Z.: A recursive method for structural learning of directed acyclic graphs. J. Mach. Learn. Res. 9, 459–483 (2008)
19. Yu, J., Smith, A., Wang, P.P., Hartemink, A.J., Jarvis, E.D.: Advances to Bayesian network inference for generating causal networks from observational biological data. Bioinformatics 20(18), 3594–3603 (2004)

Flow-Based Bayesian Estimation of Nonlinear Differential Equations for Modeling Biological Networks

Nicolas J.-B. Brunel[1] and Florence d'Alché-Buc[1,2]

[1] Laboratoire IBISC, Université d'Evry, France
[2] URA 2171, Institut Pasteur, France
{nicolas.brunel,florence.dalche}@ibisc.fr

Abstract. We consider the problem of estimating parameters and unobserved trajectories in nonlinear ordinary differential equations (ODEs) from noisy and partially observed data. We focus on a class of state-space models defined from the integration of the differential equation in the evolution equation. Within a Bayesian framework, we derive a non-sequential estimation procedure that infers the parameters and the initial condition of the ODE, taking into account that both are required to fully characterize the solution of the ODE. This point of view, new in the context of state-space models, modifies the learning problem. To evaluate the relevance of this approach, we use an Adaptive Importance Sampling in a population Monte Carlo scheme to approximate the posterior probability distribution. We compare this approach to recursive estimation via Unscented Kalman Filtering on two reverse-modeling problems in systems biology. On both problems, our method improves on classical smoothing methods used in state space models for the estimation of unobserved trajectories.

1 Introduction

1.1 Context

In recent years, there has been a growing interest in identifying complex dynamical systems in biochemistry and biology [15]. In this context, Ordinary Differential Equations (ODEs) have been widely studied for analyzing the dynamics of gene regulatory and signaling networks [11, 14]. They also appear as good candidates for the reverse-modeling task. In the present work, we consider the problem of estimating parameters and unobserved trajectories in differential equations from experimental data. Nowadays, parameter estimation in differential equations is still considered as a challenging problem when the dynamical system is only partially observed through noisy measurements and exhibit nonlinear dynamics. This is usually the case in reverse-modeling of regulatory and signaling networks [2, 16]. Some approaches address the estimation problem based on a Bayesian estimation of state-space models that integrate the ODE in the evolution equation. This framework has shown to be relevant in producing

T.M.H. Dijkstra et al. (Eds.): PRIB 2010, LNBI 6282, pp. 443–454, 2010.

efficient algorithms [20, 16, 21]. However, they suffer from two drawbacks: first they largely neglect the role of the initial condition and second, they assume the gaussianity of the posterior probability distribution of the parameters. In the present work, we are mainly interested in eliminating the first drawback by taking into account that the initial condition is a key parameter of the ODE solution. This means that we search the system parameters and the initial conditions that fit the observed data and also provide a proper solution to the ODE. Then, as a secondary contribution, we also also improve the Bayesian approach derived in [16] and in [21] by a better approximation of the posterior probability distribution.

1.2 Strategy

We first define the estimation task by introducing into the equations the flow of the ordinary differential equation. The flow of an ODE puts emphasis on the sensitivity of its solution with respect to the initial conditions. Then, we use an augmented approach that encapsulates the initial conditions and the ODEs parameters into the same augmented initial condition vector. Within this framework, the deterministic nature of the hidden process provides a non-recursive definition of hidden states from the augmented initial condition, with an integration of the ODE in the whole time interval of observation.

At this stage, we propose to address the problem with a Bayesian approach, searching for the posterior probability distribution of the augmented initial condition. The solutions previously proposed for recursive estimation in the case of nonlinear systems are based on nonlinear extensions of Kalman filtering and smoothing. We notice that procedures like Unscented Transform methods used for computing the posterior probability of the states make a strong assumption about the Gaussianity of the posterior distribution. Whereas particle filters do not make this assumptions, they do not deal correctly with deterministic processes as pointed out by the work of [13].

The idea of approximating the posterior probability by a weighted sample is computationally attractive while being a versatile approach adapted to a large variety of distributions. With respect to these considerations, we investigate the use of Monte-Carlo methods [9] for the approximation of the posterior distribution by a weighted sample built from an iterative importance sampling resampling scheme. As recently shown by [7] and [3], this approach consists in an adaptive selection of the importance distribution, which is crucial in high-dimensional sampling. The updating mechanism of the importance distribution consists in moving the population with a transition kernel (D-kernel [7]). The non-recursive estimation of the augmented initial condition is applied on two typical systems biology models: the α-pinene network [19] and the Repressilator network ([8]).

The paper is organized as follows. In section 2, we introduce the new setting of parameter estimation in terms of augmented initial condition estimation and exploit it in the context of Bayesian estimation. In section 3, we recall the main features of Population Monte-Carlo schemes and focus on an adaptive algorithm that corrects the importance distribution. Section 4 is devoted to numerical experiments. Finally, we draw a conclusion and perspectives to this work in the section 5.

2 The Initial Condition Learning Problem

2.1 Flow of an ODE and Statistical Modeling

We consider a biological dynamical system, for instance a gene regulatory network, modeled by the following ordinary differential equation:

$$\dot{x}(t) = f(t, x(t), \theta) \tag{1}$$

defined on the time interval $[0, T]$ $(T > 0)$. $x(t)$ is the state vector of dimension d: in the case of a regulatory network, it corresponds to the vector of the expression levels of d genes. f is a (time-dependent) vector field from $\mathbb{R}^d$ to $\mathbb{R}^d$, indexed by a parameter $\theta \in \Theta \subset \mathbb{R}^p$. Examples of functions f abound in the literature of systems biology [15]: Hill kinetics, law of mass equations, ...

A relevant way to characterize the differential equation under study is to define its flow $\phi_\theta : (t, x_0) \mapsto \phi_\theta(t, x_0)$ which represents the influence of $x(0) = x_0$ on the solution, i.e. $t \mapsto \phi_\theta(t, x_0)$ is the solution to (1) starting from x_0. Hence, the flow puts emphasis on the sensitivity of a solution of (1) with respect to the initial conditions. It can also be seen as a map defined in the phase space that shows how the points are transported by the vector field.

Now, let us introduce N noisy measurements, $y_n \in \mathbb{R}^m, n = 0...N-1$, that are acquired from a smooth observation function $h : R^d \to R^m$ $(m \geq 1)$ at N times $t_0 = 0 < t_1 < \ldots < t_{N-1} = T$:

$$y_n = h(\phi_\theta(t_n, x_0)) + \epsilon_n \tag{2}$$

where the noise ϵ_n is supposed to be Gaussian and homoscedastic.

If we want to fully identify the ODE, we must estimate both the parameter θ and the initial condition $x(0)$ so that the solution $\phi_{\hat{\theta}}(\cdot, \hat{x}_0)$ of the system fits the observations $y_{0:N-1} = (y_0, \ldots, y_{N-1})$. The estimation of ODE parameters by classical approaches (such as least squares [12]) is standard but gives rise to difficult global optimization problem [1]. To solve this kind of problem, variants of least square methods have been recently developed and use approximations of the solution in a spline basis (in the spirit of functional data analysis) as the generalized smoothing proposed by Ramsay et al [17], or two-step estimators [6]. When some states are hidden (typically $m < d$), the estimation (optimization step) is particularly difficult and alternative approaches have been proposed, building on the state-space model interpretation of the couple of equations (1-2). Indeed, Sitz et al. [20] first introduced a state-space model that encapsulates a differential equation in the hidden process and make use of filtering algorithms for deriving an estimate of θ. Subsequent works have exploited the same framework [16, 21, 6], but the initial condition of the system is estimated as a by-product of the filtering/smoothing steps, and the estimated states are also approximations of the solution of the ODE.

In this work, we keep the same state-space model, and we develop a Bayesian estimator which is a quite natural in state-space models, and permits the use of prior information for ameliorating the estimation. Moreover, our aim is to modify the iterative approach and to show that there is a benefit in jointly estimating θ and the initial condition x_0 in this framework. A classical evaluation of the estimated system $\dot{x}(t) = f(t, x(t), \hat{\theta})$ is to measure the quality of the fit between the true sequence $y_{0:N-1}$ and the predicted sequence $\hat{y}_n = \phi\left(t_n, (\hat{\theta}, \hat{x}_0)\right)$. Now this simple evaluation requires to know the initial value x_0 of the system, due to the one-to-one relationship between the solution of an Initial Value Problem (IVP) and an initial value x_0. Hence, despite the little interest of x_0 in general applications, it is in fact fundamental to estimate correctly x_0 in order to disentangle the influence of the parameter from the one of the initial value. Therefore, we suppose that the initial condition x_0 is unknown, so that we are also interested in its estimation. Finally, we want to estimate the augmented initial condition $z_0 = (x_0, \theta) \in R^{d+p}$ of the augmented state ODE model:

$$\begin{cases} \dot{x}(t) = f\left(t, x(t), \theta(t)\right) \\ \dot{\theta}(t) = 0 \end{cases} \tag{3}$$

with initial condition $z_0 = (x_0, \theta)$. The solution is the function $t \mapsto \phi(t, z_0)$ from $[0, T]$ to R^{p+d}. For sake of notational simplicity, we will note again this augmented ODE in R^{p+d} with the same vector field f:

$$\dot{z}(t) = f(t, z(t)) \tag{4}$$

and $z(0) = z_0$. Hence, the estimation of z_0 consists only in estimating the initial condition z_0 in (4), from $y_{0:N-1}$:

$$y_n = h(z(t_n)) + \epsilon_n$$

where we keep the notation h for the observation function from R^{p+d} to R^m $h : z = (x, \theta) \mapsto h(x)$. Now, the observed and discretized differential equation (4) fits itself in the (discrete-time) framework of state-space models in R^{p+d} with a deterministic hidden state evolution:

$$\begin{cases} z_{n+1} = z_n + \int_{t_n}^{t_{n+1}} f(\tau, z(\tau), \theta) d\tau \\ y_n \quad\; = h(z_n) + \sigma\epsilon_n \end{cases} \tag{5}$$

The state-space representation is usually exploited for deriving recursive estimation either in Maximum Likelihood approaches or in Bayesian setting as described in [5]. However, we notice an important feature of the last setting (5): the deterministic evolution of the states implies that we can compute exactly the states at each time from the initial condition (parameter) z_0, and in particular the hidden part of z_n. In equation (5), the evolution equation can be replaced by the following non-recursive definition of z_n:

$$z_n = \phi(t_n, z_0) = z_0 + \int_0^{t_n} f(\tau, z(\tau), \theta) d\tau \tag{6}$$

ϕ, as a function of the initial state z_0 is the flow of the ODE and describes the way the differential equation move the points in the phase space: the Bayesian estimation of z_0 consists in retrieving the starting point when we observe imperfectly the flow at different times.

2.2 Flow-Based Bayesian Estimation

We consider the Bayesian inference framework for the estimation of the augmented initial condition. We call Flow-based Bayesian Estimation (FBE), the Bayesian approach that consists in estimating the augmented initial condition. Since ϵ_n is Gaussian, the likelihood can be written as follows:

$$L(y_{0:N-1}; z_0) \propto \exp\left(-e(y_{0:N-1}, z_0)\right) \tag{7}$$

where $e(y_{0:N-1}, z_0) = \frac{1}{2\sigma^2}\sum_{n=0}^{N-1} \|y_n - h(\phi(t_n, z_0))\|^2$ is the classical squared error term. In the Bayesian framework, we complete the information on the parameter by a prior distribution for z_0 whose density is π_{-1}, which gives the following posterior distribution

$$\pi_{N-1}(z_0) = p(z_0|y_{0:N-1}) \propto \exp\left(-e(y_{0:N-1}, z_0)\right)\pi_{-1}(z_0) \tag{8}$$

As usual, the normalizing constant of the posterior distribution is unknown. Moreover, we have the additional computational complexity due to the absence of closed-form for the flow ϕ. Hence the Bayesian inference relies on the computation of a reliable approximation of $\pi_{N-1}(z_0)$, from which we can derive Bayesian estimators such as the posterior mean $E(Z_0|y_{0:N-1})$ or the Maximum *A Posteriori* (MAP) estimate $\arg\max_{z_0} \pi_{N-1}(z_0)$. In the MAP case, as the flow is highly nonlinear and produces wiggly likelihood functions ([17]), the direct computation of (8) is difficult and the corresponding global optimization algorithm requires intensive computations. This motivates the use of fast approximate optimization or computation of the posterior distribution, which are widely developed for non-linear state-space models. Indeed, the computation of this posterior probability can be done efficiently by recursive smoothing algorithms [5], such as Extended Kalman Filtering/Smoothing (EKF/EKS) [21], Unscented Kalman Filtering/Smoothing (UKF/UKS [20, 16] and more generally sequential Monte Carlo methods (particle filters). These classical algorithms are based on recursive computations of the filtering probabilities $p(z_n|y_{0:n})$ and several versions do exist for the computation of the smoothing probabilities $p(z_n|y_{0:N-1})$. However, in these algorithms, the initial condition is estimated as an initial state and not as a parameter of the flow. The filtering probability is even characterized by the forgetting of the initial condition as the number of observations N tends to infinity. This implies that the more data we observe, the less information we get on the parameter z_0. As a consequence, the use of refined smoothing strategies remains problematic and calls for careful adaptations ([13, 10]). Moreover, the estimated (smoothed) trajectories of the hidden states are yet not solutions of the ODE on the whole time interval $[0, T]$: the simulated trajectories may differ significantly from the smoothed trajectories (obtained by Kalman recursions or Particle filtering). Therefore, it might be preferable to turn to a non-recursive estimation of the augmented initial condition based on the non-recursive definition pointed out in (6). Taking into account the flow of the differential equation, we maximize the likelihood of exact solutions starting from different initial conditions, instead of selecting parameters admissible for describing the local transitions (based only the parameter θ).

3 Posterior Probability Estimation Using Population Monte Carlo

To test our hypothesis about the potential interest of a better estimation of the initial conditions in a Bayesian setting, we need to estimate the posterior distribution probability defined in (8). The intractability of the posterior distribution is a well-known problem in Bayesian estimation. Several general simulation methods have been developed such as Markov Chain Monte Carlo (MCMC), Importance Sampling (IS) and variants [18] are commonly used and both are well-suited to the Bayesian setting. However, one difficulty of this Monte Carlo methods is that they can be very (computationally) intensive: this is typically the case for general Hastings-Metropolis algorithms, even if some optimization can be performed. A challenging difficulty of ODE learning is that the evaluation of the likelihood is costly due to the integration of the ODE. This point motivates us to focus on importance sampling algorithms. These methods require only a "reasonable" amount of likelihood evaluations if the importance distribution is not too far from the true posterior distribution.The pitfalls of this method are well-known and are recalled in the next section, but they can be reduced by using some recently introduced adaptive schemes that we will recall.

3.1 Adaptive Importance Sampling and Population Monte-Carlo Algorithm

The principle of importance sampling is to use a Monte Carlo approximation derived thanks to a proposal (or importance) distribution q easier to simulate than π_{N-1} and to make a change of measure by introducing the weight function $w = \frac{\pi_{N-1}}{q}$:

$$E_{\pi_{N-1}}(h(Z_0)) = E_q(h(Z_0)w(Z_0)) \simeq \frac{1}{M}\sum_{i=1}^{M} h(\xi_i)w(\xi_i) \tag{9}$$

where ξ_i are i.i.d. realizations of the distribution q. Hence, the importance sampling estimators are expressed as weighted means $\frac{1}{M}\sum_{i=1}^{M}\omega_i h(\xi_i)$. Since the posterior distribution is known only up to a normalizing constant, self-normalized importance sampling estimators are rather used i.e. $\sum_{i=1}^{M}\tilde{\omega_i} h(\xi_i)$, where $\tilde{\omega_i} = \frac{\omega_i}{\sum_{i=1}^{M}\omega_i}$. The values $\tilde{\omega_i}$ are called the (normalized) importance weight, and one can interpret identity (9) as the approximation of π_{N-1} by the weighted empirical measure $\hat{\mu}_M(z) = \sum_{i=1}^{M} \tilde{\omega}_i \delta_{\xi_i}(z)$. Nevertheless, applications of IS can be very delicate as a "good" proposal distribution depends on the unknown target distribution. If the distribution q has a weak overlap with π_{N-1} (i.e. high variance of the so-called importance weights $w(\xi_i) = \omega_i$), then the IS estimators can be very poorly behaved. In that case, we have a so-called weight degeneracy, which means that all the weights vanish except on. This situation can be detected by checking the Shannon entropy of the weighted population. Another pitfall , harder to detect, is when the samples ξ_i have explored insufficiently the tails of the target distribution.

Hence a reasonable prior knowledge of the target distribution is needed to avoid these pitfalls is needed but it remains hard to have especially when dealing with posterior distribution. In order to come up with these limitations, we propose to use the Population Monte Carlo framework developed by Cappé et al [4] for deriving an adaptive Importance Sampling algorithm for dynamical systems.

Population Monte Carlo algorithm. Population Monte Carlo (PMC) is a sequential Monte Carlo method, i.e. it is an iterated Importance Sampling Resampling algorithm (ISR) which sequentially moves and re-weights a population of weighted particles $(\xi_i, \tilde{\omega}_i), i = 1, \ldots, M$. An essential feature of this algorithm is the resampling step that enables to discard particles with low weights, and to duplicate particles with high weights: this mechanism prevents then the degeneracy of the weights (i.e. all the weights vanish except one), as it is commonly used in particle filtering for instance. In all generality, a PMC scheme is defined for $t = 0, 1, \ldots, T$ and a sequence of proposal distributions q_t defined on (R^{d+p})

1. Generate $(\xi_{i,0})_{1 \leq i \leq M} \sim q_t$ (i.i.d sampling) and compute normalized weights $\tilde{\omega}_{i,t}$,
2. Resample $(\tilde{\xi}_{i,0})_{1 \leq i \leq M}$ by multinomial sampling with weights $\tilde{\omega}_{i,t}, i = 1, \ldots, M$
3. Construct q_{t+1} from $((\tilde{\xi}_{i,t'}, \tilde{\omega}_{i,t'}))_{1 \leq i \leq M, 0 \leq t' \leq t}$

The essential interest of PMC is to introduce a sequence of proposal distributions that are allowed to depend on all the past which enables to consider adaptive IS procedure based on the performance of the previous populations. PMC offers then a great versatility through the construction of the sequence of distribution q_t. In that case, the PMC estimator is still unbiased and the weights depends of step t, i.e. $w_{i,t} = \frac{\pi_{N-1}(\xi_{i,t})}{q_{i,t}(\xi_{i,t})}$. Next, we present a possible construction of a sequence of proposal distributions.

3.2 Markovian Transition and Adaptive Kernels

A simple way to randomly perturb a population is to add an independent noise to each particle $\xi_{i,t-1}$, i.e. to modify independently each particle $\xi_{i,t} = \xi_{i,t-1} + \epsilon_{i,t}$ with $\epsilon_{i,t} \sim N(0, \Sigma_t)$ (usually $\Sigma_t = \sigma_t^2 I_{d+p}$). Then, at each iteration t, we have

$\xi_{i,t} \sim N(\xi_{i,t-1}, \Sigma_t)$. General moves from $\xi_{i,t-1}$ to $\xi_{i,t}$ are described with a (Markov) transition kernel $K_{i,t}(\xi_{i,t-1}, \cdot)$. Through the resampling mechanism, particles moving in good regions are duplicated and particles moving to low credibility regions do vanish which permits a global amelioration of the population. This evolution rule described above is a simple random walk, and the mean size of the jumps is controlled by σ_t. The variance of the proposal is related to the speed at which we do move from an uninteresting region a space to an interesting one. This move is very basic, and it is interesting to propose at least several size of jumps by using a mixture of D Gaussian transition kernels: $\epsilon_{i,t} \sim \sum_{d=1}^{D} \alpha_d N(0, \Sigma_{d,t})$. With such a D-kernel, the population is moved at each iteration t at different speed $\Sigma_{d,t}$ selected with probability α_d. The D-kernel used in [4] can behave in a satisfying manner, but this algorithm is not fully adaptive as the evolution rule is not updated, whatever the success of the proposed move. Hence a better adaptive kernel is to change the move according to the survival rate of a given move. This problem of determining the weights of the mixture of kernel proposals can be seen as an estimation problem where the weights α_d used are chosen for minimizing a Kullback-Leibler divergence with an EM-like algorithm [7].

4 Experimental Results and Discussion

In this section, we compare the results provided by Unscented Kalman Smoothing (UKS) and Flow-based Bayesian Estimation using the Importance Sampling scheme (FBE-IS) and the adaptive Population Monte Carlo (FBE-PMC). We measure the quality of the approximation (estimates of the posterior covariance matrices) and also the quality of the reconstruction of the hidden states. In both cases, we consider a relatively small number of observations so that the posterior distribution is far from being approximately Gaussian. Then, we need to use approximation that can take into account multi-modality. In the cases worked out, we have used a so-called multi-start UKS based on 50 random initializations of z_0 for initiating the smoothing algorithm as described in [6]. We select the solution with the smallest quadratic error (along a trajectory) among the 50 different approximated posterior means and the corresponding posterior covariances. We first present results on a nonlinear dynamical biochemical system fully observed with noise. In this case, the state-space model reduces to the discretization of a system of ODEs, observed with some additional gaussian noise. We use this model to test the relevance of the FBE algorithm in a simple case. Then, we turn to the Repressilator which is a partially observed and nonlinear model of a gene regulatory network .

4.1 α-pinene

The α-pinene model presented here is a biochemical system of 5 interacting chemical species. It describes the isomerization of α-pinene, and the dynamics of the concentrations of the 5 species involved is described through the following time homogeneous linear ODE:

$$\begin{cases} \dot{x}_1 = -(p_1 + p_2)x_1 \\ \dot{x}_2 = p_1 x_1 \\ \dot{x}_3 = p_2 x_1 - (p_3 + p_4)x_3 + p_5 x_5 \\ \dot{x}_4 = p_3 x_3 \\ \dot{x}_5 = p_4 x_3 + p_5 x_5 \end{cases} \tag{10}$$

The evolution of the system is controlled by 5 rate constants $\theta = (p_1, \ldots, p_5)$ that we wish to estimate from noisy time series. This estimation problem is relatively classical and it has been introduced as a benchmark for the estimation of ODE, [19]. The system is completely observed and the number of observations is $N = 8$. In [19], the parameter θ has been estimated by global optimization of the least squares criterion. We use their estimate as a reference value (see reference value θ^{ref} in table 1) as it provides a good fit to the data. In this case, the situation is relatively simple as the initial condition x_0 is known and equals to $[100, 0, 0, 0, 0]^\top$ and the system is completely observed. For the Bayesian estimation, we use a non-informative uniform distribution for θ and defined on $[\theta^{ref} + 10^{-3}]$. We compare the 3 methods (UKS, FBE−IS and FBE−PMC D-kernel) only for the estimation of the parameters, see table 1, since we can set x_0 to its true value directly in FBE−IS and FBE−PMC. Finally, we use $M = 5000$ particles, and the resampled population of FBE–IS is used as the starting population for FBE–PMC. The proposal for FBE–IS is Gaussian (not centered on θ^{ref}) and is homoskedastic with standard deviation equals to 3×10^{-6}. We use $D = 7$ kernels with different variances: $\sigma_1 = 10^{-11}, \sigma_2 = 10^{-10}, \sigma_3 = 10^{-9}, \sigma_4 = 10^{-8}, \sigma_5 = 10^{-7}, \sigma_6 = 10^{-6}, \sigma_7 = 10^{-5}$. The results in table 1 show that FBE–PMC improves on the other estimates, as it is closer to the reference value, and it gives also a smaller standard deviation than UKS and FBE–IS. Moreover, results provided by FBE–PMC are more reliable than FBE–IS (or UKS), as the entropy of the PMC population $S^{PMC} = 4.3$ is bigger than the entropy of IS population $S^{IS} = 0.15$, which indicates that FBE–PMC enables to avoid degeneracy of the population, and the weights are well scattered. Finally, at iteration $t = 20$ of FBE–PMC, it remains only 1 component with variance 10^{-9} and a population with entropy equals to 4.5. Finally, the reconstructed trajectories obtained by UKS and FBEPMC show that better fitting and predicting model is provided by FBEPMC, thanks to the initial value parametrization.

4.2 An Example of a Partially Observed System: The Repressilator Network in *E. coli*

The Repressilator network was proposed in 2000 [8] to describe sustained oscillations observed in a small system in the bacterium *E. coli*, composed of three genes that code for 3 proteins. The first repressor protein, LacI from *E. coli*, inhibits the transcription of the second repressor gene, TetR, from the tetracycline-resistance transposon Tn10, whose protein product in turn inhibits the expression of a third

Table 1. Estimated parameter values with UKS, FBE–IS and FBE–PMC with standard deviation

	Reference ($\times 10^{-5}$)	UKS ($\times 10^{-5}$)	FBE–IS ($\times 10^{-5}$)	FBE–PMC (D-Kernels) ($\times 10^{-5}$)
p_1	5.9259	3.66 ± 5.6	$5.98 \pm 1.3 \times 10^{-2}$	$5.93 \pm 4.7 \times 10^{-2}$
p_2	2.9634	2.5 ± 4.8	$2.92 \pm 1.3 \times 10^{-2}$	$2.96 \pm 5 \times 10^{-2}$
p_3	2.0473	1.78 ± 20.4	$2.05 \pm 5.69 \times 10^{-2}$	$2.06 \pm 2 \times 10^{-2}$
p_4	27.4490	27.3 ± 31.1	$26.7 \pm 5.69 \times 10^{-2}$	$27.89 \pm 10 \times 10^{-2}$
p_5	3.9980	4.24 ± 26	$3.53 \pm 13.1 \times 10^{-2}$	$4.11 \pm 5.2 \times 10^{-2}$
$\left\| \hat{\theta} - \theta^{ref} \right\|$	0	2.3×10^{-5}	8.2×10^{-6}	4.5×10^{-6}

gene, CI from λ phage. Finally, CI inhibits LacI expression, completing the cycle. Hill kinetics are used to model the dynamics. For sake of simplicity in notations, x_1, x_2, x_3 denote respectively the expression of genes Lacl, TetR1, Cl and x_4, x_5, x_6 the concentrations of corresponding proteins. The network evolution is described by the following ODE:

$$\begin{cases} \dot{x_1} = v_1^{max} \frac{k_{12}^n}{k_{12}^n + x_5^n} - k_1^{mRNA} x_1 \\ \dot{x_2} = v_2^{max} \frac{k_{23}^n}{k_{23}^n + x_6^n} - k_2^{mRNA} x_2 \\ \dot{x_3} = v_3^{max} \frac{k_{31}^n}{k_{31}^n + x_4^n} - k_3^{mRNA} x_3 \\ \dot{x_4} = k_1 x_1 - k_1^{protein} x_4 \\ \dot{x_5} = k_2 x_2 - k_2^{protein} x_5 \\ \dot{x_6} = k_3 x_3 - k_3^{protein} x_6 \end{cases} \tag{11}$$

In the simulations, the RNA concentrations x_1, x_2, x_3 are supposed to be observed through a noisy measurement process (modeled by a Gaussian with $\sigma = 3$) while the proteins concentrations x_4, x_5, x_6 are not measured. The initial values are also supposed to be unknown, and need then to be estimated. The true parameter values and initial conditions are available in tables (2, 3) respectively. We use a Gaussian distribution centered at z_0^{true} as a prior, and the proposal distribution q is a Gaussian distribution with a shifted mean ($z_0^{true} + 2$) and homoskedastic covariance (with standard deviation = 5). For measuring the performance of the 3 estimators, we perform a Monte Carlo study with $N^{MC} = 100$ independent replicates (and we use populations of size $M = 1000$). The mean results in tables (2,3) show that FBE–PMC is unbiased and gives reliable confidence results whereas UKS provide significantly different estimates, and important standard deviation. In the case of Importance Sampling the mean estimates are correct, but the standard deviation are very small. This come from the weights degeneracy of FBE–IS: indeed, in 84% of simulations, a single particle has a weight greater than 90% (the mean entropy of the weights of the IS population is 0.12). This is not the case for FBE–PMC which avoids this curse with $T = 10$ steps starting from the population used by Importance Sampling (the mean entropy is 2.24). In particular from table 3 one can see that FBE–PMC enables to gives more credible values for the hidden states, and guarantees also that the corresponding solution with estimated z_0 is close to the data.

Table 2. Estimated Parameters using UKS, FBE–IS and FBE–PMC (D-kernel) $T = 25$ observations. Average Means and Standard Deviations computed on 100 samples.

Parameter	True Parameter	UKS	FBE–IS	FBE–PMC
v_1^{max}	150	147.3 ± 0.9	150.2 ± 0.09	150.0 ± 0.46
v_2^{max}	80	81.9 ± 1.7	80.7 ± 0.49	80.2 ± 0.66
v_3^{max}	100	102.2 ± 1.7	100.7 ± 0.25	100.1 ± 0.91
k_1	50	53.0 ± 0.9	50.7 ± 0.05	50.1 ± 0.35
k_2	30	37.1 ± 0.94	30.9 ± 0.08	29.9 ± 0.38
k_3	40	47.6 ± 0.8	40.72 ± 0.03	40.0 ± 0.36

Table 3. Estimated Initial Conditions of Hidden States using UKS, FBE–IS and FBE–PMC (D-kernel) approach $T = 25$ observations. Average Means and Standard Deviations computed on 100 samples.

Parameter	True Parameter	UKS estimation	FBE–IS	FBE–PMC
$p_1(0)$	1	97.8 ± 5.9	3.11 ± 0.21	2.86 ± 0.09
$p_2(0)$	2	143.6 ± 3.0	3.83 ± 0.21	3.51 ± 0.10
$p_3(0)$	3	148.5 ± 8.6	4.76 ± 0.17	4.75 ± 0.27

5 Conclusion and Perspective

We have proposed to learn both the initial condition and the parameters in such a way that they convey a proper solution of the ODE. As in biological or biochemical experiments, the initial condition vector can be fully observed, we turn the ODE estimation problem into a state-space model estimation task where the only parameter to estimate is an augmented initial condition. A Bayesian approach to this problem, called FBE, has been derived using an Importance Sampling schme (FBE–IS) and a Population Monte Carlo scheme (FBE–PMC) for the approximation of the posterior probability. The FBE–PMC approach overcomes classical limitations of standard estimation methods in state-space models. The versatility of the PMC schemes gives new estimation methods, based on the learning of proposal distribution q_t that permits a better exploration of the space. The engineering of proposal distributions adapted to the dynamical systems remains quite unexplored and links with particle filters might be pointed out. Finally, a promising research direction for reverse-modelling of biological networks is to combine the augmented initial condition estimation with the graph structure estimation in the Bayesian framework.

Acknowledgement. This work was supported by the ANR Projects GD2GS (2006–2009) and ODESSA (2009–2012).

References

[1] Rodriguez-Fernandez, M., Egea, J.A., Banga, J.R.: Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems. BMC Bioinformatics 7(483) (2006)

[2] Calderhead, B., Girolami, M., Lawrence, N.D.: Accelerating bayesian inference over nonlinear differential equations with gaussian processes. In: Koller, D., Schuurmans, D., Bengio, Y., Bottou, L. (eds.) Advances in Neural Information Processing Systems, vol. 21, pp. 217–224. MIT Press, Cambridge (2009)

[3] Cappé, O., Douc, R., Guillin, A., Marin, J.M., Robert, C.P.: Adaptive importance sampling in general mixture classes. Statistics and Computing 18(4), 447–459 (2008)

[4] Cappé, O., Guillin, A., Marin, J.M., Robert, C.P.: Population monte carlo. Journal of Computational and Graphical Statistics 13(4), 907–929 (2004)

[5] Cappé, O., Moulines, E., Rydén, T.: Inference in Hidden Markov Models. Springer, Heidelberg (2005)

[6] d'Alché-Buc, F., Brunel, N.J.-B.: Learning and inference in computational systems biology. In: Estimation of Parametric Nonlinear ODEs for Biological Networks Identification. MIT Press, Cambridge (2010)

[7] Douc, R., Guillin, A., Marin, J.M., Robert, C.: Convergence of adaptive mixtures of importance sampling schemes. Annals of Statistics 35(1), 420–448 (2007)

[8] Elowitz, M., Leibler, S.: A synthetic oscillatory network of transcriptional regulators. Nature 403, 335–338 (2000)

[9] Gentle, J.E., Hardle, W., Mori, Y.: Handbook of computational statistics: concepts and methods. Springer, Heidelberg (2004)

[10] Ionides, E., Breto, C., King, A.: Inference for nonlinear dynamical systems. Proceedings of the National Academy of Sciences 103, 18438–18443 (2006)

[11] de Jong, H.: Modeling and simulation of genetic regulatory systems: A literature review. Journal of Computational Biology 9(1), 67–103 (2002)

[12] Li, Z., Osborne, M.R., Prvan, T.: Parameter estimation of ordinary differential equations. IMA Journal of Numerical Analysis 25, 264–285 (2005)

[13] Liu, J., West, M.: Combined parameter and state estimation in simulation-based filtering. In: Doucet, A., de Freitas, N., Gordon, N. (eds.) Sequential Monte Carlo Methods in Practice, pp. 197–217. Springer, Heidelberg (2001)

[14] Mendes, P.: Learning and inference in computational systems biology. In: Comparative Assessment of Parameter Estimation and Inference Methods. MIT Press, Cambridge (2010)

[15] Lawrence, N., Girolami, M., Rattray, M., Sanguinetti, G.: Learning and Inference in Computational Systems Biology. MIT Press, Cambridge (2010)

[16] Quach, M., Brunel, N., d'Alché-Buc, F.: Estimating parameters and hidden variables in non-linear state-space models based on odes for biological networks inference. Bioinformatics 23(23), 3209–3216 (2007)

[17] Ramsay, J.O., Hooker, G., Campbell, D., Cao, J.: Parameter estimation for differential equations: A generalized smoothing approach. Journal of the Royal Statistical Society, Series B 69, 741–796 (2007)

[18] Robert, C.P., Casella, G.: Monte Carlo Statistical Methods. Springer, Heidelberg (2004)

[19] Rodriguez-Fernandez, M., Egea, J.A., Banga, J.R.: Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems. BMC Bioinformatics 7(483) (2006)

[20] Sitz, A., Schwarz, U., Kurths, J., Voss, H.: Estimation of parameters and unobserved components for nonlinear systems from noisy time series. Physical review E 66, 16210 (2002)

[21] Sun, X., Jin, L., Xiong, M.: Extended kalman filter for estimation of parameters in nonlinear state-space models of biochemical networks. PLoS ONE 3(11), e3758+ (2008)

Author Index