Mehmet Koyutürk
Shankar Subramaniam
Ananth Grama *Editors*

Functional Coherence of Molecular Networks in Bioinformatics

Springer

Functional Coherence of Molecular Networks in Bioinformatics

Mehmet Koyutürk • Shankar Subramaniam
Ananth Grama
Editors

Functional Coherence of Molecular Networks in Bioinformatics

Springer

Editors

Mehmet Koyutürk
Department of Electrical Engineering
and Computer Science
Case Western Reserve University
10900 Euclid Ave.
Cleveland Ohio 44106
USA
mxk331@case.edu

Shankar Subramaniam
Department of Bioengineering
University of California
San Diego
9500 Gilman Dr.
La Jolla California 92093-0412
USA
shankar@sdsc.edu

Ananth Grama
Department of Computer Science
Purdue University
305 North University Street
West Lafayette 47907-2107
Indiana
USA
ayg@cs.purdue.edu

ISBN 978-1-4614-0319-7 e-ISBN 978-1-4614-0320-3
DOI 10.1007/978-1-4614-0320-3
Springer New York Dordrecht Heidelberg London

Library of Congress Control Number: 2011937231

© Springer Science+Business Media, LLC 2012
All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.
The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

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

Preface

The past decade has seen tremendous advances in our understanding of the basic mechanisms that underlie living organisms. These advances have been motivated by our ability to gather large amounts of data relating to the state of living systems (chemical compositions, dynamic fluxes, binding states, etc.), modeling in ways that lend themselves to powerful analyses techniques, and associated algorithms and software. In contrast to traditional approaches that take a deconstructive view of biological processes (scaling up from atomistic and molecular levels), systems biology studies the organization and emergent properties of interacting units (typically biomolecules). This poses profound challenges, as well as exciting opportunities for new discoveries. This edited volume focuses on the computational challenges associated with systems-level modeling of biological function and explores how functional relationships among biomolecules are manifested in biological networks.

Systems-level (network) models of biological systems typically rely on graphs with nodes corresponding to biological entities and edges corresponding to interrelationships among these entities. Constructing such models from data, reasoning from these models, characterizing their organization and function, and relating them to the genotype/phenotype, pose complex modeling and algorithmic challenges. While one may be tempted to view these as traditional graph algorithms, well studied in computer science literature, the unique characteristics of biological interaction data necessitates development of novel models and methods. These characteristics include: (1) incomplete and noisy datasets, (2) highly skewed distributions, (3) need for establishing statistical validity, (4) incorporating elements of space, time, and scale, and (5) relating across disparate abstractions.

While there is tremendous ongoing research and development activity in this area, there is an established core of results that provides the basis for future development. This book provides a comprehensive overview of the state-of-the-art, as it relates to modeling and analysis of living systems as interacting biological units. By the nature of the underlying processes, the book intersects broad subdisciplines within biology, statistics, and computer science. It addresses methods for data collection and curation, model inference, statistically grounded analyses, algorithms, software and frameworks, and validation. It also motivates a number of open problems, which provide excellent avenues for continuing efforts in the area.

The book is the result of significant efforts on part of the contributing authors. The editors would like to thank all of the authors for their outstanding contributions.

Cleveland, OH, USA — Mehmet Koyutürk
West Lafayette, IN, USA — Ananth Grama
La Jolla, CA, USA — Shankar Subramaniam

Contents

1 **Introduction to Network Biology** 1
Mehmet Koyutürk, Shankar Subramaniam, and Ananth Grama

2 **Topological Characteristics of Molecular Networks** 15
Tijana Milenković and Nataša Pržulj

3 **Function Annotation in Gene Networks** 49
Petko Bogdanov, Kathy Macropol, and Ambuj K. Singh

4 **Proteome Network Emulating Models** 69
Phuong Dao, Fereydoun Hormozdiari, Iman Hajirasouliha, Martin Ester, and S. Cenk Sahinalp

5 **Biological Network Alignment** 97
Shahin Mohammadi and Ananth Grama

6 **Pattern Mining Across Many Massive Biological Networks** 137
Wenyuan Li, Haiyan Hu, Yu Huang, Haifeng Li, Michael R. Mehan, Juan Nunez-Iglesias, Min Xu, Xifeng Yan, and Xianghong Jasmine Zhou

7 **Molecular Networks and Complex Diseases** 171
Mehmet Koyutürk, Sinan Erten, Salim A. Chowdhury, Rod K. Nibbe, and Mark R. Chance

8 **Moving Toward Genome-Scale Kinetic Models: The Mass Action Stoichiometric Simulation Approach** 201
Aarash Bordbar and Bernhard Ø. Palsson

Index 221

Contributors

Petko Bogdanov Department of Computer Science, University of California, Santa Barbara, CA 93106, USA, petko@cs.ucsb.edu

Aarash Bordbar University of California, San Diego, CA 92093-5004, USA, aabordba@ucsd.edu

Mark R. Chance Case Western Reserve University, Cleveland, OH 44106-3808, USA, mrc16@case.edu

Salim A. Chowdhury Carnegie-Mellon University, Pittsburgh, PA 15213, USA, sachowdh@andrew.cmu.edu

Phuong Dao School of Computing Science, Simon Fraser University, 8888 University Drive, Burnaby, B.C., V5A 1S6 Canada, pdao@cs.sfu.ca

Sinan Erten Case Western Reserve University, Cleveland, OH 44106-3808, USA, mse10@case.edu

Martin Ester Simon Fraser University, Burnaby, B.C., V5A 1S6 Canada, ester@cs.sfu.ca

Ananth Grama Department of Computer Science, Purdue University, 305 N. University Street, West Lafayette, IN 47907-2107, USA, ayg@purdue.edu

Iman Hajirasouliha Simon Fraser University, Burnaby, B.C., V5A 1S6 Canada, imanh@cs.sfu.ca

Fereydoun Hormozdiari Simon Fraser University, Burnaby, B.C., V5A 1S6 Canada, fhormozd@cs.sfu.ca

Haiyan Hu University of Central Florida, 4000 Central Florida Boulevard Orlando, FL 32816, USA, haihu@cs.ucf.edu

Yu Huang University of Southern California, Los Angeles, CA 90007-2439, USA, yuhuang@usc.edu

Mehmet Koyutürk Department of Electrical Engineering and Computer Science, Case Western Reserve University, 10900 Euclid Ave., Cleveland, OH 44106, USA, mxk331@case.edu

Haifeng Li Motorola Labs, Los Angeles, CA 90013, USA, haifengl@usc.edu

Wenyuan Li University of Southern California, Los Angeles, CA 90007-2439, USA, wel@usc.edu

Kathy Macropol University of California, Santa Barbara, Santa Barbara, CA 93106, USA, kpm@cs.ucsb.edu

Michael R. Mehan University of Southern California, Los Angeles, CA 90007-2439, USA, rielmeha@usc.edu

Tijana Milenković Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA, tmilenko@nd.edu

Shahin Mohammadi Purdue University, Lafayette, IN 47906, USA, mohammas@purdue.edu

Rod K. Nibbe Case Western Reserve University, Cleveland, OH 44106-3808, USA, rkn6@case.edu

Juan Nunez-Iglesias University of Southern California, Los Angeles, CA 90007-2439, USA, nunezigl@usc.edu

Bernhard Ø. Palsson Department of Bioengineering, University of California – San Diego, 9500 Gilman Dr., La Jolla, CA, USA, palsson@ucsd.edu

Nataša Pržulj Imperial College London, Exhibition Rd, London SW7 2AZ, UK, natasha@imperial.ac.uk

S. Cenk Sahinalp Simon Fraser University, Burnaby, B.C., V5A 1S6 Canada, cenk@cs.sfu.ca

Ambuj K. Singh University of California, Santa Barbara, CA 93106, USA, ambuj@cs.ucsb.edu

Shankar Subramaniam University of California, San Diego, CA 92093-5004, USA, shankar@sdsc.edu

Min Xu University of Southern California, Los Angeles, CA 90007-2439, USA, mxu@usc.edu

Xifeng Yan University of California, Santa Barbara, CA 93106, USA, xyan@cs.ucsb.edu

Xianghong Jasmine Zhou Program in Computational Biology, Department of Biological Sciences, University of Southern California, Los Angeles, CA 90089, USA, xjzhou@usc.edu

Chapter 1
Introduction to Network Biology

Mehmet Koyutürk, Shankar Subramaniam, and Ananth Grama

Abstract In the study of biological systems, complex interactions among biomolecules are often abstracted using network models. Molecular networks describe the organization of various biological processes, including cellular signaling, metabolism, and genetic regulation. These networks are commonly used for a system-level understanding and analysis of biological function. Indeed, research and development efforts focused on such analyses have grown significantly over the past few years. This edited volume provides a detailed description of current models and methods focused on functional characterization of biological networks. In this introductory chapter, we provide a broad overview of molecular network models and their relation to biological function.

1 Systems Biology

At the core of our understanding of biological processes and underlying systems, is a characterization of the function and interactions of their constituent parts. In medical sciences, understanding the origin of functional anomalies holds the key to effective diagnosis, treatment, and prognosis. In genetics, functional annotation of genetic variability uncovers complex relationships between genotype and phenotype. In evolutionary biology, functional differences between diverse organisms highlight the evolutionary mechanisms that underlie the complexity of biological systems. With the successful completion of the human genome project and recent technological advances in biological data collection, it has become possible to study function from a systems perspective. Today, systems biology is established as a fundamental

M. Koyutürk (✉)
Department of Electrical Engineering and Computer Science, Case Western Reserve University,
10900 Euclid Ave., Cleveland, OH 44106, USA
e-mail: mxk331@case.edu

M. Koyutürk et al. (eds.), *Functional Coherence of Molecular Networks in Bioinformatics*, DOI 10.1007/978-1-4614-0320-3_1,
© Springer Science+Business Media, LLC 2012

interdisciplinary science that focuses on detailed studies of the complex mechanisms that orchestrate the interactions between various biomolecules that compose life.

Systems biology is defined as the study of an organism as a "system." Often, this study is restricted to a single cell – aiming to characterize the structure and dynamics of cellular *function* as a whole, in contrast to studying the structure and function of individual components in isolation [27]. To this end, systems biology focuses on understanding the organization of biological systems in terms of the interconnectivity of cellular components (e.g., who interacts with whom?), as well as the dynamics of these interactions (e.g., to what extent do these components interact?). In doing so, systems biology takes into account the key characteristics of complex systems, including emergence, robustness, and modularity.

In modeling complex systems, there is often a trade-off between the level of detail and the scale of the model [10]. In other words, to comprehensively understand the dynamics of a biological process, it is desirable to understand the dynamic properties of its components, as a function of time, and in relation to each other (e.g., location of atoms, quantity of molecules). This requires a highly resolved model of the structure and function. On the other hand, the complexity of a cell requires modeling at a much coarser level. Scaling beyond a single cell to an entire organism requires a higher level of abstraction. The number of protein-coding genes in the human genome is estimated to be around 25 K; multiple levels of control, including transcriptional control, alternative splicing, and post-translational modifications greatly expand the space of possible states for individual molecules. Interactions among these molecules further extend the space of cellular states combinatorially. According to a recent survey, the human body is composed of about 10^{14} cells of more than 200 different types [46]. The role of suitable models as we rise up through this hierarchy is clearly indicated. While advances in biotechnology have enabled generation of large amounts of data on biological systems, modeling these systems with the desired level of detail and at the desired resolution remains a challenging task. Network models offer a critically useful abstraction in this regard; they provide comprehensive models of the wiring of cellular systems, while at the same time serving as a template for more detailed dynamic models [51].

2 Molecular Interaction Networks

Cellular systems are organized hierarchically, from individual molecules (e.g., genes, mRNA, proteins, metabolites) to large scale molecular pathways [43]. In this organization, interactions among molecules play an important role; interacting molecules are organized into functional modules (aggregates that participate in similar function), which in turn interact with each other to drive larger scale biological processes [17]. Comprehensive maps of the interactions among biomolecules provide an integrated view of the cell. While these interactions underlie dynamic orchestration of cellular tasks as a system [27], information regarding the dynamics of these interactions is often not readily available. The past decade has witnessed significant efforts aimed at modeling, identifying, organizing, and analyzing cellular interactions. These efforts, grounded in significant advances in our understanding of molecular biology, are enabled by high-throughput data collection and acquisition techniques that are used to interrogate the states and interactions of biomolecules at multiple levels.

In computational analysis of cellular interactions, graph models are commonly used [29]. The wiring of biomolecules through pairwise, as well as multiway interactions is abstracted using different network models. In general, network models represent molecules by nodes, and their interactions by edges (links). These links indicate interactions in different forms, including physical binding, regulatory interactions, genetic interactions, and computationally predicted functional association [60]. Common abstractions for molecular interactions include protein–protein interaction (PPI) networks, gene regulatory networks, and metabolic networks. The interactions modeled using these abstractions are closely interrelated and the underlying components of the network cannot be viewed in isolation from each other. However, individual models provide a simplified view of different modes of interaction, facilitating efficient organization and analysis of these interactions.

2.1 Protein–Protein Interaction Networks

An important class of molecular interaction data is in the form of PPIs. Knowledge of these interactions provides an experimental basis for understanding modular organization of cells, as well as useful information for predicting the biological function of individual proteins [59]. High throughput screening methods such as yeast two-hybrid (Y2H) [22], mass spectrometry (MS) [20], and tandem affinity purification (TAP) [14], provide large amounts of data on the *interactome* of an increasing number of species. This data is organized into several public databases, including Database of Interacting Proteins (DIP) [69], BioGrid [62], and Human Protein Reference Database (HPRD) [48]. Availability of these databases allows a broad range of scientists to analyze PPIs from a global network perspective.

The commonly used Y2H screening identifies interactions between pairs of proteins by exploiting the modularity of the activating and binding domains of eukaryotic transcription factors [22]. Namely, in Y2H, the activating and binding domains of a specific transcription factor are separated. Subsequently, each of the activating and binding domains is fused to one of the two (prey and bait) proteins being assayed. Finally, the expression of a target (referred to as the reporter gene) of the transcription factor is measured. Since the reporter gene would not be expressed when the activating and binding domains of the transcription factor are separated, the high expression of the reporter gene indicates an interaction between prey and bait proteins. TAP, on the other hand, identifies interactions between a single bait protein and multiple other proteins [14]. This is achieved by tagging the protein of interest (referred to as the bait protein) and introducing it to the host. Once the bait

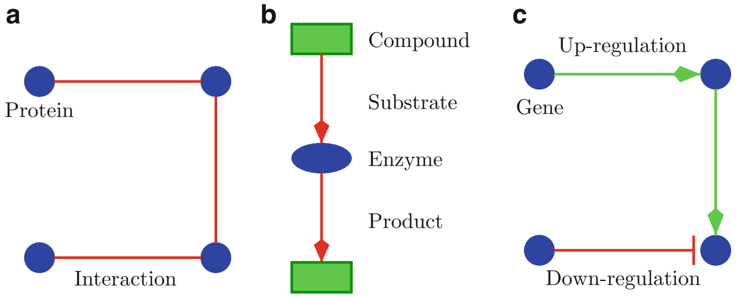


Fig. 1.1 Graph models for molecular interactions: (**a**) Protein interaction networks, (**b**) Metabolic pathways, (**c**) Gene regulatory networks

protein is retrieved, the proteins attached to the bait protein are identified using MS and these proteins are considered as interacting partners of the bait protein.

Established PPI network models assume binary interactions between pairs of proteins, which is naturally descriptive of the outcome of Y2H screening. These pairwise interactions are modeled using simple undirected graphs in which nodes represent proteins and an edge between two nodes represents the interaction between the corresponding proteins, as shown in Fig. 1.1a. On the other hand, multiple interactions identified by TAP are either modeled as hypergraphs in which edges are replaced by hyperedges [42], or inserted into the binary network model as either: (1) a star network around the bait protein (spoke model) or (2) a clique of all proteins retrieved by the bait protein, including itself (matrix model) [55].

An important limitation of PPIs derived from high-throughput techniques is their incomplete and noisy nature [11]. Furthermore, these interactions only represent a snapshot of the dynamical organization of proteins in the cell; many interactions may be transient and condition-dependent, while some others are permanent [23]. Currently available PPI datasets are also highly prone to ascertainment bias. Furthermore, high-throughput screening does not reveal the structural bases of identified PPIs. Protein interactions may arise from interactions between structural domains, which have relatively large interfaces, or domain-small polypeptide stretch interactions with relatively low affinity, therefore, less likely to be detected [3]. Since knowledge of structural bases of PPIs is useful in understanding their functional bases, many computational methods have been developed to infer domain–domain interactions (DDIs) from PPIs [16, 33, 53]. Comprehensive comparison of PPI and DDI networks has shown that DDI networks provide more reliable information on the functional relationships among biomolecules, as compared to PPI networks [59].

PPIs are also inferred using various computational techniques. These methods use different sources of experimental data to assess the likelihood of functional association between a pair of proteins. Common computational techniques used for predicting PPIs include phylogenetic profiling [26, 47] and analysis of gene expression [23, 63], based on the premise that interacting proteins are likely to have coevolved or be coexpressed, since their cooperative task would require existence of both proteins. Since protein interaction data obtained from high-throughput screening is highly error-prone [65, 66], it is common to combine several experimental and computational sources of interaction data to obtain a reliable set of putative interactions. Such consolidated interaction networks are modeled using weighted graphs, where edge weights represent the likelihood of interaction between proteins, estimated using various statistical models and techniques [4, 34]. Machine learning models are also useful in consolidating PPI networks [61].

2.2 Metabolic Networks

Metabolic networks comprise a historically well-studied abstraction for biological networks. They characterize interconnected chains of chemical reactions that occur in a living organism to maintain life. These chains of reactions can synthesize larger molecules from simpler molecules (anabolism) or break down molecules into simpler ones to release energy (catabolism). Traditionally, metabolic networks are dissected into specific metabolic pathways, based on the products of each group of reactions underlying a metabolic process. With recent developments in the application of computational methods to cell biology, there have been successful attempts at modeling, synthesizing [24], and organizing metabolic pathways into public databases such as KEGG [41], MetaCyc [31], and EMP [56]. Enzyme Nomenclature provides a unified view of metabolic reactions across species [64].

Metabolic pathways are chains of reactions, in which reactions are linked to each other by chemical compounds (metabolites) through product–substrate relationships. A natural mathematical model for metabolic pathways is a directed hypergraph in which each node corresponds to a compound, and each hyperedge corresponds to a reaction (or equivalently enzyme) [32]. The direction of a pin of a hyperedge indicates whether the compound is a substrate or product of the reaction. This model is illustrated in Fig. 1.1b. It is possible to replace this model by a simpler directed graph if, for instance, we are only interested in relationships between enzymes. In such a model, enzymes correspond to nodes of the graph and a directed edge from one enzyme to another indicates that a product of the first enzyme is a substrate of the second. Indeed, metabolic pathways are represented in terms of various binary relations in KEGG [15].

While the pathway representation is useful in cataloging and organizing metabolic processes, it is important to note that metabolic pathways are interconnected. Indeed, global network representation of metabolic networks enables the study of metabolic network dynamics from a systems perspective. Metabolic flux models based on steady-state assumptions prove invaluable in this regard [45]. While Michelis–Menten kinetics provides a mathematical foundation for the kinetics of individual reactions and catalyzing enzymes, these reactions form a system together, which is difficult to analyze in terms of the parameters of Michelis–Menten kinetics. However, stoichiometric models provide a linear algebraic framework for dynamic analysis of metabolic networks as a system, based on the assumption that metabolic concentration changes are much slower than reaction kinetics.

2.3 Gene Regulatory Networks

The cell adapts to its environment by recognition and transduction of a broad range of environmental signals, which in turn activate response mechanisms by regulating the expression of proteins that take part in the corresponding processes. Mechanisms of cellular signaling and genetic regulation also play key roles in cellular communication in multicellular organisms, including developmental processes. A fundamental challenge in systems biology is, therefore, to reconstruct networks that describe cellular signaling and regulation, with a view to deriving maps of interconnectivity and functional relationships between molecules.

At the transcriptional level, gene expression is regulated through interaction of regulatory proteins (e.g., transcription factors) with the DNA at specific locations. The combinatorial relationship between transcription factors and their target genes are organized into transcriptional regulatory networks, providing qualitative models of genetic regulation at the level of transcription [2]. Transcriptional networks can be reconstructed by detecting genes whose expression is coregulated, finding common motifs in the neighborhood of these genes as potential candidates for their promoters, and subsequently identifying specific protein–DNA interactions [35].

Besides transcriptional regulation, gene and protein expression is regulated in other phases, including post-transcription [7], translation [19], and post-translation control. While identification of all such regulatory mechanisms is a challenging task, correlations between expression levels of genes provide valuable information regarding regulatory interactions that extend beyond transcriptional regulation. Gene regulatory networks, also referred to as genetic networks, provide an abstract model of the regulatory effects of genes on each other, represented as directed (and often annotated) interactions among two or more genes [18].

A simple and commonly used model for gene regulatory networks is the Boolean network model [1]. In this model, the expression of each gene in the network is represented by a binary variable and the regulatory effect of the genes on a particular gene is represented as a Boolean function. In the basic representation of this model, nodes correspond to genes and a directed edge from one gene to the other represents the regulatory effect of the first gene on the second. Here, edges are labeled by the mode of regulation, which may be one of up-, down-, or dual-regulation. This model is illustrated in Fig. 1.1c.

Boolean networks provide simple, yet useful models of causal relationships in the cell. They can be surprisingly powerful in predicting cellular behavior in various contexts. However, they do not account for many important factors, including the quantitative and asynchronous nature of cellular signaling and regulation, as well as variables that are not measured. Bayesian networks utilize stochasticity to account for such factors, which are otherwise intractable [12]. Bayesian networks represent the expression of a gene as a random variable and characterize the relationship between a gene and its regulators in terms of the conditional probability distribution of this random variable with respect to the expression of regulators.

2.4 Other Abstractions

There exist various other abstractions for modeling molecular interactions. These include signal transduction pathways, which model the mechanisms for the cell to receive, process, and respond to information through signal transfer between proteins [9] and gene coexpression networks, which pack relations in complex expression patterns into pairwise associations between genes [63]. In the Molecule Pages database [37], proteins involved in cell signaling are represented in various states and transitions between these states, as an important step in abstracting cellular processes via state diagrams and eventually modeling the cell as a state machine.

3 Molecular Networks and Biological Function

Graph-theoretic modeling of biological networks provides a framework for the solution to various problems associated with understanding biological function [49]. Computational approaches that facilitate extraction of organized and annotated functional information from molecular interaction networks range from simple queries to more sophisticated analysis tasks.

Analysis of graph-theoretical properties of molecular networks reveal many properties of the networks that hint on functional relationships among multiple proteins. Fundamental observations on the relationship between basic graph-theoretical features and functional coherence of molecular networks include the following:

- It has been repeatedly observed that functionally related (e.g., proteins in the same pathway, proteins with similar molecular function, products of genes that are implicated in similar diseases) are likely to reside close to each other in molecular networks. Traditionally, the "distance" between two biomolecules in a network of interactions is measured by the minimum number of interactions that separate a given pair of proteins [50]. This measure is further extended to random-walk based models, which capture the functional relationship between proteins more accurately [59].
- It has been shown that the network centrality of a biomolecule (e.g., the number of interactions) is correlated with its essentiality [6]. Furthermore, the distribution of such measures as network degree or clustering coefficient also provides insights on the robustness of the network [67].

- Functionally correlated groups of biomolecules generally manifest themselves as a densely connected sub-network in a network of molecular interactions [5].

More sophisticated computational analyses target identification of patterns that exhibit certain interesting or unusual; hence, potentially functionally relevant characteristics (e.g., in terms of frequency, density, or conservation), based on the expectation that such unusual patterns reveal underlying functional requirements and/or evolutionary pressure. Such analysis techniques include the following:

- Graph clustering targets identification of dense subgraphs in the network and is commonly used for identification of functional modules and complexes [5, 30]. These algorithms are based on the notion that a group of functionally-related entities are likely to densely interact with each other while being somewhat separated from the rest of the network [54].
- Hierarchical decomposition methods rely on the observation that organization of cellular processes can be modeled using hierarchical modularity [52]. These methods use hierarchical clustering algorithms for identification of functional modules [13].
- Motif finding is based on identification of specific topological motifs that are observed significantly more often than they would be observed at random in a network of interactions. These algorithms reveal common regulatory motifs and coherent interaction patterns as putative building blocks of biological networks. They also provide insights into the functional topology of interaction networks, facilitating compact modeling and reverse engineering of these networks [7, 38, 68].
- Inferring function of individual proteins and assigning complex memberships based on proximity, topological similarity, or other more detailed network characteristics provides a useful computational tool for extracting information from interaction data [4, 36, 59].
- Comparative network analysis aims to extract evolutionary information from molecular interaction networks [57]. Comparison of networks across multiple species provides understanding of conservation and divergence of the modularity of cellular processes in an evolutionary framework for systems biology [39] and facilitates projection of functional, structural, and modular annotation for model organisms onto a diverse set of species[25, 28, 58].
- Integration of molecular network data with other omic datasets sheds light on the mechanistic bases of phenotypic differences. In particular, in the context of several diseases, molecular networks are successfully used to prioritize disease genes, identify network signatures of disease, and improve classification of phenotype [35, 40].

It should also be noted that combinatorial abstractions in computational network analysis often overlook the dynamics of the cellular interactions and provide a simplified picture of the organization. Consequently, for accurate modeling, simulation, and engineering of cellular systems, it is necessary to combine these combinatorial models and the information gained from analysis of such models with dynamic analysis techniques that target understanding of how a system behaves over time under various conditions [27].

4 Functional Coherence of Molecular Networks

The chapters in this volume provide a detailed review of the relationship between biological function and graph-theoretical characteristics of molecular networks. These chapters are organized into three parts: (a) function, (b) evolution, and (c) dynamics.

The next two chapters of this volume are devoted to the relationship between molecular network topology and biological function. In Chap. 2, Milenković and Pržulj investigate biological function from the perspective of molecular network topology. They provide an overview of computational approaches to assessing the meaning of network topology in relation to functional organization of biological systems and provide surprising results on how simple wiring patterns can play a key role in orchestration of cellular processes. In Chap. 3, Bogdanov et al. provide a detailed review of computational methods that aim to infer function of individual molecules based on molecular network data. These methods use principles that are supported by various lines of empirical evidence. These principles range from the observation that functionally associated proteins are likely to interact with each other, to the observation that interacting partners of functionally associated proteins are also likely to interact with each other.

The two subsequent chapters focus on functional evolution of molecular networks. In Chap. 4, Dao et al. discuss computational models for molecular network evolution. These models provide significant insights into the origins of observed functional characteristics of molecular networks and highlight key evolutionary pathways into emergent properties of biological systems. Building on these models that provide evolutionary insights into functional coherence of molecular networks, Mohammadi and Grama provide a detailed review of computational methods for comparative network analysis in Chap. 5. These methods include algorithms for comparison of pairs of networks for identification of conserved interaction patterns, as well as multiple network alignment and identification of orthologous proteins based on network topology.

The last three chapters of the volume focus on network dynamics and phenotype. In Chap. 6, Li et al. describe several algorithms for identifying common, as well as differential patterns on multiple molecular networks, and show how these patterns can be used for functional inference and genotype to phenotype mapping. In Chap. 7, Koyutürk et al. provide a detailed review of network-based algorithms for identification of disease-associated genes, proteins, as well as network signatures of diseases. These algorithms often utilize molecular networks in conjunction of other omic data sources, including sequence and expression data. Finally, in Chap. 8, Bordbar and Palsson introduce mass action stoichiometric simulation, with a view to providing genome-scale kinetic models for metabolic networks.

References

1. Tatsuya Akutsu, Satoru MIYANO, and Satoru KUHARA. Identification of genetic networks from a small number of gene expression patterns under the boolean network model. In *PSB*, pages 17–28, 1999.
2. Uri Alon. *An Introduction to Systems Biology: Design Principles of Biological Circuits:*. Chapman & Hall/CRC, 2006.
3. Patrick Aloy and Robert B. Russell. Structural systems biology: modelling protein interactions. *Nature Reviews Molecular Cell Biology*, 7(3):188–197, March 2006.
4. S. Ashtana, O. D. King, F. D. Gibbons, and F. P. Roth. Predicting protein complex membership using probabilistic network reliability. *Genome Research*, 14:1170–1175, 2004.
5. G. D. Bader and C. W. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4(2), 2003.
6. A. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.
7. J. Berg and M. Lassig. Local graph alignment and motif search in biological networks. *PNAS*, 101(41):14689–14694, 2004.
8. Annamaria Bevilacqua, Maria Cristina Ceriani, Sergio Capaccioli, and Angelo Nicolin. Post-transcriptional regulation of gene expression by degradation of messenger RNAs. *Journal of Cellular Physiology*, 195(3):356–372, 2003.
9. U. S. Bhalla and R. Iyengar. Emergent properties of networks of biological signaling pathways. *Science*, 283(5400):381–387, 1999.
10. Stefan Bornholdt. Less is more in modeling large genetic networks. *Science*, 310(5747): 449–451, 2005.
11. Eric de Silva, Thomas Thorne, Piers J. Ingram, Ino Agrafioti, Jonathan Swire, Carsten Wiuf, and Michael P. H. Stumpf. The effects of incomplete protein interaction data on structural and evolutionary inferences. *BMC Biol.*, 4:39+, 2006.
12. Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7(3):601–620, August 2000.
13. J. Gagneur, R. Krause, T. Bouwmeester, and G. Casari. Modular decomposition of protein-protein interaction networks. *Genome Biology*, 5:R7, 2004.
14. A. C. Gavin, M. Bösche, R. Krause, P. Grandi, M. Marzioch, and A. Bauer. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868): 141–147, 2002.
15. S. Goto, H. Bono, H. Ogata, W. Fujibuchi, K. Sato, and M. Kanehisa. Organizing and computing metabolic pathway data in terms of binary relations. In *Proceedings of Pacific Symposium on Biocomputing*, pages 175–186, 1997.
16. Katia S. Guimaraes and Teresa M. Przytycka. Interrogating domain-domain interactions with parsimony based approaches. *BMC Bioinformatics*, 9:171+, March 2008.
17. L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray. From molecular to modular cell biology. *Nature*, 402:C47–C51, 1999.
18. J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins. Computational studies of gene regulatory networks: In numero molecular biology. *Nature Reviews Genetics*, 2:268–279, 2001.
19. Alan G. Hinnebusch. Evidence for translational regulation of the activator of general amino acid control in yeast. *PNAS*, 81(20):6442–6446, 1984.
20. Y Ho, A Gruhler, A Heilbut, GD Bader, L Moore, and SL Adams. Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature*, 415:180–183, 2002.
21. Trey Ideker and Roded Sharan. Protein networks in disease. *Genome Research*, 18(4):644–652, April 2008.
22. T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *PNAS*, 98(8):4569–4574, 2001.

23. R. Jansen, D. Greenbaum, and M. Gerstein. Relating whole-genome expression data with protein-protein interactions. *Genome Research*, 12(1):37–46, 2002.
24. P. D. Karp and S. M. Paley. Representations of metabolic knowledge: pathways. In *Proceedings of 2nd International Conference on Intelligent Systems for Molecular Biology (ISMB'94)*, pages 203–211, 1994.
25. B. P. Kelley, B. Yuan, F. Lewitter, R. Sharan, B. R. Stockwell, and T. Ideker. PathBLAST: A tool for aligment of protein interaction networks. *Nucleic Acids Research*, 32:W83–W88, 2004.
26. Y. Kim, M. Koyutürk, U. Topkara, A. Grama, and S. Subramaniam. Inferring functional information from domain co-evolution. *Bioinformatics*, 22(1):40–49, 2006.
27. H. Kitano. Systems biology: A brief overview. *Science*, 295(5560):1662–1664, 2002.
28. M. Koyutürk, A. Grama, and W. Szpankowski. Pairwise local alignment of protein interaction networks guided by models of evolution. *Proceedings of 9th International Conference on Research in Computational Molecular Biology (RECOMB'05)*, LNCS 3500:48–65, 2005.
29. Mehmet Koyutürk. Algorithmic and analytical methods in network biology. *WIREs Systems Biology and Medicine*, 2(3):277–292, May 2010.
30. Mehmet Koyutürk, Wojciech Szpankowski, and Ananth Grama. Assessing significance of connectivity and conservation in protein interaction networks. *Journal of Computational Biology*, 14(6):747–764, 2007.
31. C. J. Krieger, P. Zhang, L. A. Mueller, A. Wang, S. Paley, M. Arnaud, J. Pick, S. Y. Rhee, and P. D. Karp. MetaCyc: A multiorganism database of metabolic pathways and enzymes. *Nucleic Acids Research*, 32(1):438–442, 2004.
32. L. Krishnamurthy, J. Nadeau, G. Özsoyoğlu, M. Özsoyoğlu, G. Schaeffer, M. Taşan, and W. Xu. Pathways database system: an integrated system for biological pathways. *Bioinformatics*, 19(8):930–937, 2003.
33. Hyunju Lee, Minghua Deng, Fengzhu Sun, and Ting Chen. An integrated approach to the prediction of domain-domain interactions. *BMC Bioinformatics*, 7(269), 2006.
34. I. Lee, S. V. Date, A. T. Adai, and E. M. Marcotte. A probabilistic functional network of yeast genes. *Science*, 306(5701):1555–1558, 2004.
35. T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J. B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young. Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, 298(5594):799–804, 2002.
36. Stanley Letovsky and Simon Kasif. Predicting protein function from protein/protein interaction data: a probabilistic approach. In *Bioinformatics Supplement on 11th International Conference on Intelligent Systems for Molecular Biology (ISMB'03)*, pages 197–204, 2003.
37. Joshua Li, Yuhong Ning, Warren Hedley, Brian Saunders, Yongsheng Chen, Nicole Tindill, Timo Hannay, and Shankar Subramaniam. The molecule pages database. *Nature*, 420: 716–717, 2002.
38. E. Y. Lotem, S. Sattath, N. Kashtan, S. Itzkovitz, R. Milo, R. Y. Pinter, U. Alon, and H. Margalit. Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction. *PNAS*, 101(16):5934–5939, 2004.
39. Monica Medina. Genomes, phylogeny, and evolutionary systems biology. Technical Report LBNL-57355, Lawrence Berkeley National Laboratory, 2005.
40. R.K. Nibbe, S.A. Chowdhury, M. Koyutürk, R. Ewing, and M.R. Chance. Protein-protein interaction networks and sub-networks in the biology of disease. *WIREs Systems Biology and Medicine*, page 10.1002/wsbm.121, 2011.
41. H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, and M. Kanehisa. KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 27:29–34, 1999.
42. F. Olken. Biopathways and protein interaction databases. A lecture in *Bioinformatics Tools for Comparative Genomics: A Short Course*, May 2003.
43. Z. N. Oltvai and A. L. Barabási. Life's complexity pyramid. *Science*, 298:763–764, 2002.

44. Jayesh Pandey, Mehmet Koyutürk, Shankar Subramaniam, and Ananth Grama. Functional coherence in domain interaction networks. *Bioinformatics*, 24:i28–i34, August 2008.
45. J. A. Papin, N. D. Price, S. J. Wiback, D. A. Fell, and B. O. Palsson. Metabolic pathways in the post-genome era. *Trends Biochem Sci*, 28(5):250–258, May 2003.
46. Jason A. Papin, Tony Hunter, Bernhard O. Palsson, and Shankar Subramaniam. Reconstruction of cellular signalling networks and analysis of their properties. *Nature Reviews Molecular Cell Biology*, 6(2):99–111, February 2005.
47. M. Pellegrini, E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *PNAS*, 96(8):4285–4288, 1999.
48. Prasad, Renu Goel, Kumaran Kandasamy, Shivakumar Keerthikumar, Sameer Kumar, Suresh Mathivanan, Deepthi Telikicherla, Rajesh Raju, Beema Shafreen, Abhilash Venugopal, Lavanya Balakrishnan, Arivusudar Marimuthu, Sutopa Banerjee, Devi Somanathan, Aimy Sebastian, Sandhya Rani, Somak Ray, Kishore, Sashi Kanth, Mukhtar Ahmed, Manoj Kashyap, Riaz Mohmood, Y. L. Ramachandra, V. Krishna, Abdul Rahiman, Sujatha Mohan, Prathibha Ranganathan, Subhashri Ramabadran, Raghothama Chaerkady, and Akhilesh Pandey. Human protein reference database–2009 update. 37:D767–D772+, 2009.
49. N. Pržulj. Graph theory analysis of protein-protein interactions. In I. Jurisica and D. Wigle, editors, *Knowledge Discovery in Proteomics*. CRC Press, 2004.
50. N. Pržulj, D. A. Wigle, and I. Jurisica. Functional topology in a network of protein interactions. *Bioinformatics*, 20(3):340–348, 2004.
51. Teresa M. Przytycka, Mona Singh, and Donna K. Slonim. Toward the dynamic interactome: it's about time. *Briefings in Bioinformatics*, 11(1):15–29, January 2010.
52. E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A. L. Barabási. Hierarchical organization of modularity in metabolic networks. *Science*, 297:1551–1555, 2002.
53. Robert Riley, Christopher Lee, Chiara Sabatti, and David Eisenberg. Inferring protein domain interactions from databases of interacting proteins. *Genome biology*, 6(10), 2005.
54. A. W. Rives and T. Galitski. Modular organization of cellular networks. *PNAS*, 100(3): 1128–1133, 2003.
55. D. Scholtens, M. Vidal, and R. Gentleman. Local modeling of global interactome networks. *Bioinformatics*, 21(17):3548–3557, 2005.
56. E. Selkov, S. Basmanova, T. Gaasterland, I. Goryanin, Y. Gretchkin, N. Maltsev, V. Nenashev, R. Overbeek, E. Panyushkina, L. Pronevitch, E. Selkov, and I. Yunus. The metabolic pathway collection from EMP: The enzymes and metabolic pathways database. *Nucleic Acids Research*, 24(1):26–28, 1996.
57. R. Sharan and T. Ideker. Modeling cellular machinery through biological network comparison. *Nature Biotechnology*, 24(4):427–433, 2006.
58. R. Sharan, S. Suthram, R. .M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R. M. Karp, and T. Ideker. Conserved patterns of protein interaction in multiple species. *PNAS*, 102(6): 1974–1979, 2005.
59. Roded Sharan, Igor Ulitsky, and Ron Shamir. Network-based prediction of protein function. *Mol Syst Biol*, 3, March 2007.
60. Benjamin A. Shoemaker and Anna R. Panchenko. Deciphering protein-protein interactions. part i. experimental techniques and databases. *PLoS computational biology*, 3(3):e42+, March 2007.
61. Balaji S. Srinivasan, Nigam H. Shah, Jason A. Flannick, Eduardo Abeliuk, Antal F. Novak, and Serafim Batzoglou. Current progress in network research: toward reference networks for key model organisms. *BRIEFINGS IN BIOINFORMATICS*, 8(5), September 2007.
62. C. Stark, B. J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. BioGRID: a general repository for interaction datasets. *Nucleic acids research*, 34(Database issue): D535–D539, January 2006.
63. Joshua M. Stuart, Eran Segal, Daphne Koller, and Stuart K. Kim. A gene-coexpression network for global discovery of conserved genetic modules. *Science*, 302(5643):249–255, 2003.

64. Keith Tipton and Sinéad Boyce. History of the enzyme nomenclature system. *Bioinformatics*, 16(1):34–40, January 2000.
65. B. Titz, M. Schlesner, and P. Uetz. What do we learn from high-throughput protein interaction data? *Expert Review of Proteomics*, 1(1):111–121, 2004.
66. Peter Uetz, Loic Giot, Gerard Cagney, Traci A. Mansfield, Richard S. Judson, James R. Knight, Daniel Lockshon, Vaibhav Narayan, Maithreyan Srinivasan, Pascale Pochart, Alia Qureshi-Emili, Ying Li, Brian Godwin, Diana Conover, Theodore Kalbfleisch, Govindan Vijayadamodar, Meijia Yang, Mark Johnston, Stanley Fields, and Jonathan M. Rothberg. A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, 403(6770):623–627, 2000.
67. S. Wuchty. Evolution and topology in the yeast protein interaction network. *Genome Research*, 14:1310–1314, 2004.
68. S. Wuchty, Z. N. Oltvai, and A. L. Barabási. Evolutionary conservation of motif constituents in the yeast protein interaction network. *Nature Genetics*, 35(2):176–179, 2003.
69. I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S. Kim, and D. Eisenberg. DIP: The database of interacting proteins. a research tool for studying cellular networks of protein interactions. *Nucleic Acids Research*, 30:303–305, 2002.

Chapter 2
Topological Characteristics of Molecular Networks

Tijana Milenković and Nataša Pržulj

Abstract We present currently available computational methods for graph-theoretic analysis, modeling, and comparison of biological networks. Biological network research is still in its infancy, since the current data is of low quality, and since the existing methods for their analyses are relatively crude, owing to the computational intractability of many graph theoretic problems. Nonetheless, the field has already provided valuable insights into biological function, evolution, and disease. Further systems-level analyses of cellular inter-connectedness have an enormous potential to lead to new interesting biological discoveries and give novel insights into organizational principles of life and therapeutics, thus potentially having huge impacts on public health. The impact of the field of biological network research is likely to increase with the growth of available biological network data of high quality, as well as with improvements of network analysis and modeling methods. The field is likely to stay at the forefront of scientific research in the years to come.

1 Biological Networks: Motivation, Data Sets, and Challenges

The definition of a *network* (also called a *graph*) is simple: it is a set of objects, called *nodes*, and connections between the objects, called *links* or *edges*. Networks are invaluable models for better understanding of complex systems and they have been used to describe, model, and analyze an enormous array of real-world phenomena in many research domains, including physical systems such as electrical power grids and communication networks, social systems such as networks of friendships or corporate and political hierarchies, or software systems such as call graphs or expression and syntax trees. The field of computational and systems

T. Milenković (✉)
Department of Computer Science and Engineering, University of Notre Dame,
Notre Dame, IN 46556, USA
e-mail: tmilenko@nd.edu

M. Koyutürk et al. (eds.), *Functional Coherence of Molecular Networks in Bioinformatics*, DOI 10.1007/978-1-4614-0320-3_2,
© Springer Science+Business Media, LLC 2012

biology is no exception: biological networks provide an organized and elegant framework to study and model complex events that emerge from interactions among the individual constituents of the cell.

Different types of biological networks exist, depending on the type of a biological phenomenon that they model. Nodes in biological networks represent biomolecules such as genes, proteins, or metabolites, and edges connecting the nodes indicate functional, physical, or chemical interactions between the corresponding biomolecules. An example are *protein–protein interaction (PPI) networks*, in which nodes correspond to proteins and edges exist between pairs of nodes if the corresponding proteins physically bind to each other (Fig. 2.1a). When all proteins in a cell are considered, the resulting networks are quite large, containing thousands of proteins and tens of thousands of interactions even for model organisms, such as baker's yeast *Saccharomyces cerevisiae* (Fig. 2.1b). In addition to PPI networks, many other biological phenomena have been modeled with graphs, including transcriptional regulation, cell signaling, functional associations between genes (e.g., synthetic lethality), metabolism, neuronal synaptic connections, protein structures, and brain activity. Studying biological networks at these various granularities could provide valuable insights about inner working of cells and lead to important discoveries about complex diseases. Deep understanding of these networks is one of the ultimate challenges of computational and systems biology [113, 114].

We have been witnessing the exponential growth of the amounts of available biological network data, along with the development of computational approaches for studying and modeling these data. High-throughput screens for interaction detection, such as yeast two-hybrid (Y2H) assays [39, 46, 55, 75, 108, 119, 123, 128], affinity purification coupled to mass spectrometry (AP/MS) [44, 45, 53, 67], genome-wide chromatin immunoprecipitation, correlated m-RNA expression, and genetic (synthetic-lethal) and suppressor networks [20, 127], have yielded partial networks for many model organisms [45, 46, 50, 55, 67, 75, 98, 127, 128] and humans [108, 123], as well as for bacterial [72, 95, 106] and viral [17, 129, 134] pathogens. Numerous biological network datasets are now publicly available in several databases, including Saccharomyces Genome Database (SGD),[1] the Database of Interacting Proteins (DIP),[2] Human Protein Reference Database (HPRD),[3] and the Biological General Repository for Interaction Datasets (BioGRID).[4]

Biological network research is an interdisciplinary and integral part of computational and systems biology that offers many interesting and important opportunities for biological and computational scientists. Systems-level analyses of inter-connectedness of the cell are promising to provide new insights into organizational principles of life, evolution, disease, and therapeutics, thus potentially having huge impacts on public health. Unlike genetic sequence research, biological

[1] http://www.yeastgenome.org/.

[2] http://dip.doe-mbi.ucla.edu/.

[3] http://www.hprd.org/.

[4] http://www.thebiogrid.org/.

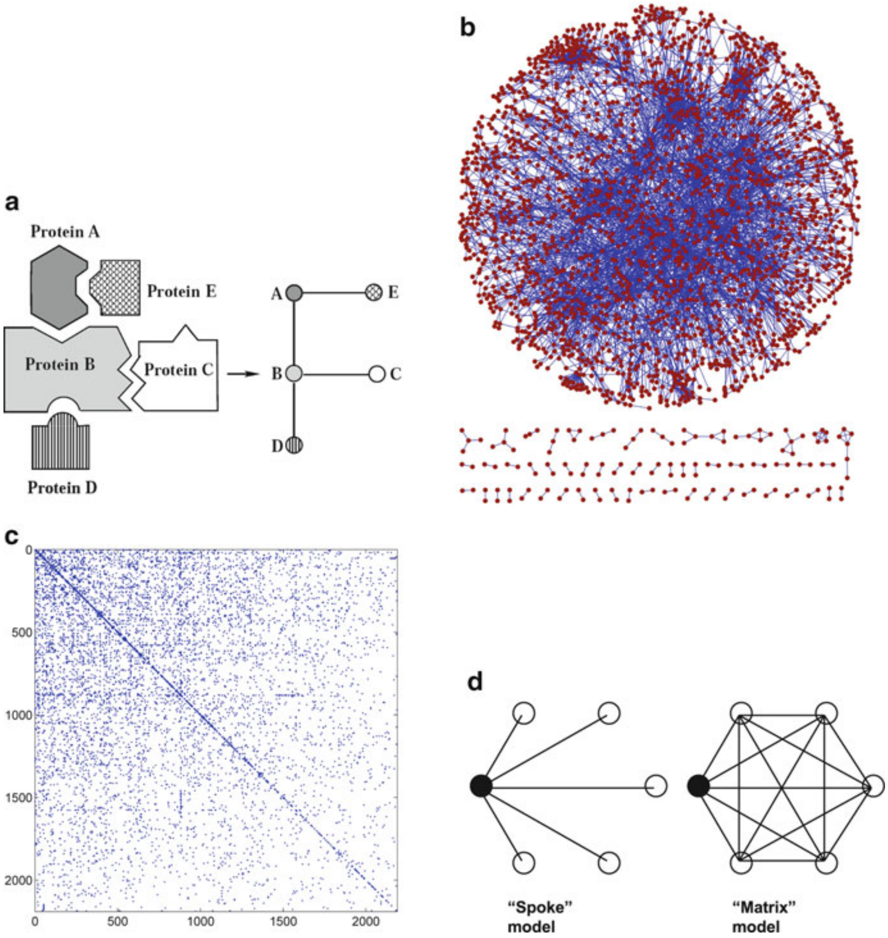


Fig. 2.1 (**a**) A schematic representation of a protein–protein interaction (PPI) network. (**b**) Baker's yeast PPI network downloaded from Database of Interacting Proteins (DIP) [142]. (**c**) The adjacency matrix of the same network illustrating its sparsity. (**d**) An illustration of the "spoke" and "matrix" models for defining PPIs in a pull-down experiment. The *black node* corresponds to the bait, while *white nodes* correspond to preys

network research is still in its infancy. We have only begun collecting the biological network data and we can hardly even describe the data mathematically, much less understand it theoretically. Nonetheless, deep biological understanding has already been obtained by studying biological networks. The ultimate expectation is that network data will be at least as useful as the sequence data in uncovering new biology. Since biological networks are very large and complex (e.g., see Fig. 2.1b), it is not possible to understand them without computational analyses and modeling.

The emerging field of network biology faces considerable challenges. Currently available biological network data sets are noisy and largely incomplete (Fig. 2.1c) with some parts being more dense than others, owing to the following. First, experimental techniques are limited, as they are capable of extracting only samples of interactions that exist in the cell. Second, the data sets incorporate biases introduced by humans during data collection [19, 22, 23, 48, 49, 124, 135, 141]. An example is a bias introduced by collecting more data in parts of networks that are relevant for human disease due to increased interest and availability of funding. Another example is a bias of adding noise to the data by using the "spoke" or the "matrix" model to define interactions between proteins identified in a pull-down AP/MS experiment. In the "spoke" model, interactions are assumed between the bait and all of the preys, while in the "matrix" model, additional interactions are assumed between all preys as well (Fig. 2.1c). Clearly, the spoke model introduces fewer false positives than the matrix model, but it can miss true interactions. Despite the incompleteness and noisiness of currently available biological network data, the scientific community has begun analyzing and modeling the networks, since they represent a rich source of biological information. This has led to interesting but sometimes controversial discoveries [21, 27, 33–35, 56–58, 62, 122, 126]. The controversies often resulted from a lack of understanding of the sampling properties of the data, as well as from the use of computational techniques sensitive to noise [23, 49, 124].

Moreover, analyzing, modeling, and comparing biological network data is nontrivial not only because of the low quality of currently available biological network data, but also due to provable computational intractability of many graph theoretic problems. Modeling of biological networks is of particular importance, since a good network model that reproduces a real-world phenomenon well can help us understand the laws governing the phenomenon, and only with the help of such laws we can make new predictions about the phenomenon. Finding a good network model requires comparing the data with model networks, but exact network comparisons are computationally infeasible. Hence, we must rely on approximate solutions resulting from heuristic algorithms. However, even if computational intractability was not a problem, arriving to exact graph theoretic solutions would be inappropriate in biology due to biological variation. Hence, we want our methods intentionally to be heuristic.

In this chapter, we introduce network analysis and modeling methods that are commonly applied to biological networks. We mainly focus on PPI networks, since it is the proteins that carry out almost all biological processes. However, the same methods can easily be applied to other biological networks. This chapter is organized as follows. In Sect. 2, we describe the main computational concepts related to network analysis and introduce measures of network topology. In Sect. 3, we present the commonly used network models and describe their biological applications. In Sect. 4, we introduce existing approaches for network alignment and explain how they can be used to transfer biological knowledge between species. In Sect. 5, we present approaches linking network topology with biological function and disease. Finally, in Sect. 6, we present some open problems in biological network research and future expectations of the field, and we give some concluding remarks.

2 Network Analysis: Measures of Network Structure

Network analyses require contrasting different networks and finding their topological (or structural) similarities and differences. For example, it would be useful to compare networks of different species, since a lot is often known about a network of a model organism, but very little about networks of other organisms. Also, it would be useful to evaluate the fit of different models to the data. Hence, network comparisons are essential for any network data analysis. However, exact comparisons of large networks are computationally infeasible, owing to the computational intractability of the underlying subgraph isomorphism problem, which asks if a graph exists as an exact subgraph of another graph. More formally, a graph *isomorphism* between two networks is a node bijection preserving the node adjacency relation [139]. If two networks G and H are given as input, determining whether G contains a subgraph isomorphic to H has mathematically been proven to be NP-complete (meaning that no efficient way of finding a solution exists), since it includes problems such as Hamiltonian path, Hamiltonian cycle, and the maximum clique as special cases [43]. However, if graph G on n_G nodes is input and graph H on n_H nodes is fixed, then the subgraph isomorphism can be tested in polynomial time, $O(n_H! \cdot n_H^2 \cdot \binom{n_G}{n_H})$, simply by iterating through all subsets of n_H nodes of G. Nonetheless, such exhaustive searches are computationally infeasible for large biological (and other real-world) networks and hence approximate, heuristic approaches are sought. Furthermore, as mentioned above, due to biological variation and noise in the biological network data (i.e., missing edges, false edges, or both [133]), subgraph isomorphism would be inappropriate in biology even if it was computationally feasible. For this reason, we want our network comparison methods intentionally to be more flexible, or approximate.

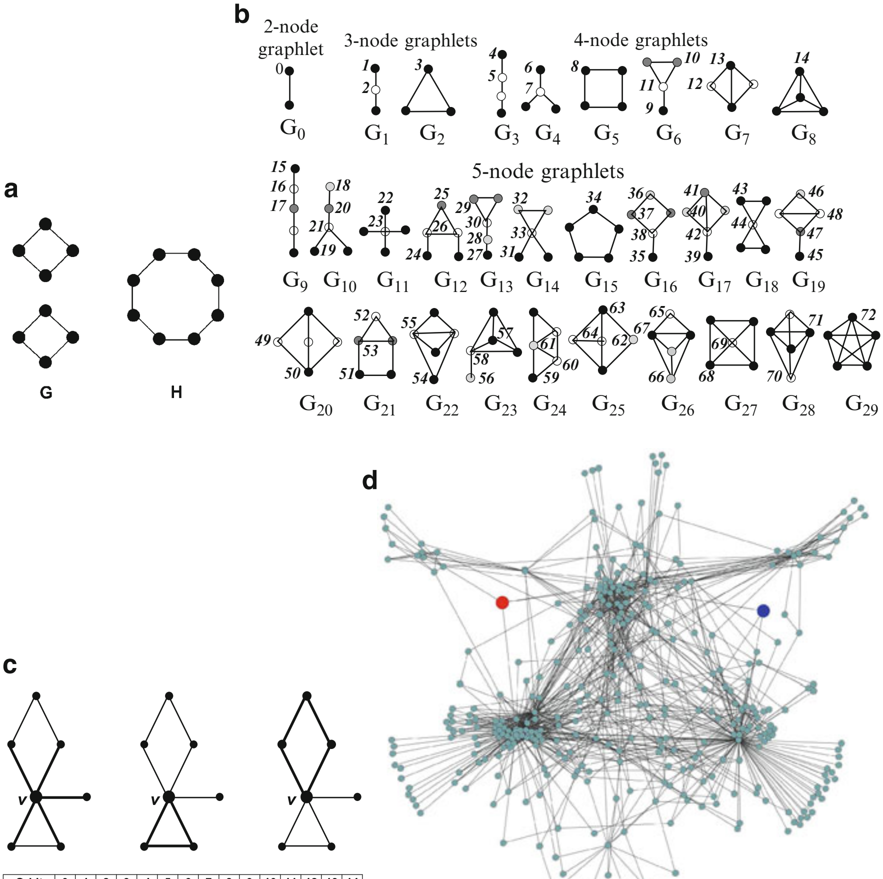
Easily computable approximate measures of network topology commonly used to characterize a network and compare different networks are referred to as *network properties*. Network properties are used, for example, to compare the structure of real-world networks with the structure of model networks. Based on such comparisons, network models have been proposed for cellular (and other real) networks if their properties fit the properties of cellular networks (Sect. 3). Network properties have traditionally been divided into two main groups: top-down macroscopic *global network properties* (Sect. 2.1) and bottom-up microscopic *local network properties* (Sect. 2.2). Additionally, various forms of node *centralities* have been proposed, characterizing the "topological importance" of a node based on its position in the network (Sect. 2.3).

2.1 Global Network Properties

Global properties of a network give an overall view of the network. They are conceptually and computationally easy and thus, they have been extensively studied in biological networks. The most widely used global network properties are the *degree distribution, clustering coefficient, clustering spectrum, network diameter*, and *spectrum of shortest path lengths* [90]. They are defined as follows.

The *degree* of a node is the number of edges that the node touches. For example, in the network presented in Fig. 2.1a, nodes C, D, and E have degree 1, node A has degree 2, and node B has degree 3. The *degree distribution* of a network is the distribution of degrees of all nodes in the network, measuring the percentage of nodes in the network having degree k, for all degrees k in the network. Equivalently, it is the probability that a randomly selected node of the network has degree k. This probability is commonly denoted by $P(k)$. Many biological networks have skewed, asymmetric degree distributions with a tail that follows a "power-law" given by the following formula: $P(k) \sim k^{-\gamma}, \gamma > 0$. Networks with such degree distributions are referred to as "scale-free" [7], and in these networks, the largest percentage of nodes has degree 1, much smaller percentage of nodes has degree 2, and so forth, but there exists a small number of high-degree nodes called "hubs." The *clustering coefficient* of a node v can be viewed as the probability that two neighbors of the node v are connected; neighbors of v are nodes that share an edge with v. Specifically, the clustering coefficient of node v is the percentage of edges that exist amongst v's neighbors out of the maximum possible number of edges amongst the neighbors. The clustering coefficient of a network is simply the average of clustering coefficients of all of its nodes. It measures the tendency of the network to form highly interconnected regions called clusters. Clearly, it is always between 0 and 1. The *clustering spectrum* of a network is the distribution of average clustering coefficients of degree k nodes over all degrees k in the network. The smallest number of links that have to be traversed to get from a node v to a node u in a network is called the *distance* between nodes v and u and a path through the network that achieves this distance is called the *shortest path* between nodes v and u. The maximum of shortest path lengths over all pairs of nodes in a network is the network's *maximum diameter* that describes how "far spread" the network is. The average of shortest path lengths over all pairs of nodes in a network is called the network's *average diameter*. PPI and other biological and real networks have high clustering coefficients compared to completely random networks, as well as small average diameters (of the order of $O(\log n)$, where n is the number of nodes in a network); this is called the *small-world* property [138].

However, global network properties might not be constraining enough to capture at a detailed level complex topological characteristics of large (biological) networks. For example, networks with exactly the same value of a global network property can have very different structure affecting their function [74, 100]. Figure 2.2a presents such an example: network G, consisting of two 4-node cycles, and network H, consisting of one 8-node cycle, have the same number of nodes and edges, as well as the same degree distribution (each node in both networks has degree 2) and clustering spectrum (each node in both networks has the clustering coefficient of 0); however, their network structure is clearly very different. The same holds for other global network properties [100]. Hence, more constraining measures of network



Orbit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
GDV (*v*)	5	2	9	1	0	6	0	7	1	0	0	3	0	0	0

Fig. 2.2 (**a**) Two networks, *G* and *H*, that have the same size, degree distribution, and clustering spectrum, but different topology. (**b**) All the connected graphs on 2–5 nodes. When appearing as induced subgraphs of a larger network, we call them *graphlets*. They contain 73 topologically unique node types, called "automorphism orbits." In a particular graphlet, nodes belonging to the same orbit are of the same shade [99]. (**c**) An illustration of the "Graphlet Degree Vector" (GDV) of node *v*. Graphlet G_0 in panel (**b**) is just an edge, and the degree of a node historically defines how man edges it touches. We generalize the degree (*left*) to a 73-component GDV that counts how many times a node touches each of the 73 automorphism orbits, such as a *triangle* (*middle*), or a *square* (*right*). For illustration purposes, the GDV of node *v* is presented in the table for orbits 0–14: *v* is touched by 5 edges (orbit 0), end-nodes of 2 graphlets G_1 (orbit 1), etc. (**d**) Illustration of GDV-similarity measure: 2-deep network neighborhoods of proteins DDX6 (*red node*) and ZNF384 (*blue node*) that have high GDV-similarity of 97%

structure are needed for proper network comparisons. Furthermore, since currently available biological networks are incomplete, their global network properties do not tell us much about the structure of the entire real-world networks. Instead, they describe the structure of "localized" sampled network parts that were produced by biased experimental techniques for interaction detection [23, 49, 124] (see the discussion in Sect. 1). Thus, global statistics on incomplete real-world networks may be biased as well and even misleading with respect to the currently unknown complete networks. Since certain local neighborhoods of biological networks are well-studied (e.g., network regions relevant for human disease), bottom-up *local* network properties might be more appropriate to analyze and model the well-studied network parts.

2.2 Local Network Properties

Local network properties include network motifs and graphlets [83,86,99,100,117]. Analogous to sequence motifs, *network motifs* are defined as subgraphs that appear in a network at frequencies much higher than those in randomized networks [85, 86, 117]. Equivalently, anti-motifs are subgraphs that appear in a network at frequencies much lower than those in randomized networks. Network motifs are partial subgraphs. A *partial subgraph* H of a network G is a subgraph whose nodes and edges belong to G. An *induced subgraph* H of G is a subgraph of G on subset $V(H)$ of the set of nodes $V(G)$, such that edges $E(H)$ of H consists of *all* edges of G that connect nodes of $V(H)$. The following example illustrates the difference between partial and induced subgraphs: a 3-node path is a partial subgraph of a triangle, since all nodes and edges of a 3-node path belong to a triangle. However, a 3-node path is not an induced subgraph of a triangle, since not all edges that exist between the tree nodes in a triangle are present in a 3-node path; the triangle has a single induced subgraph – the triangle. Hence, induced subgraphs are more topologically constraining than partial subgraphs. Beside being partial subgraphs, network motifs raise several additional issues. First, motif discovery requires comparing real-world networks with random model networks. However, it is not clear which random network model should be used for this purpose [5]. Using an inadequate model may identify as over-represented subgraphs that otherwise would not have been identified as motifs. (We discuss network models in more detail in Sect. 3.) Second, when focusing on discovery and analysis of network motifs, one ignores subgraphs with "average" frequencies. However, it is as important to understand why certain network structures appear at average in the data as it is to understand why some structures are under- and over-represented, if we are to get a complete understanding of the underlying cellular processes. Despite these drawbacks, network motifs have been very useful for finding functional building blocks of transcriptional regulation networks, as well as for differentiating between different types of real-world networks [2, 60, 61, 85, 86, 117]. Also, as partial subgraphs, network motifs are appropriate for studying biological networks, since not all interactions in real biological networks need to concurrently occur in a cell, while they are all present simultaneously in their network representations that we study.

Unlike network motifs, *graphlets* are small *induced* subgraphs in a network (Fig. 2.2b) [26, 82, 83, 99, 100]. Graphlet-based approaches have been proposed that count the frequencies of occurrences of *all* graphlets in a network, not only over-represented ones. As such, these approaches are free from the biases that motif-based approaches have: graphlets are induced subgraphs and they can be identified without the need to use any random graph model. That is, graphlets do not need to be over-represented in a data network and this, along with being induced, distinguishes them from network motifs. (Note that whenever the structure of a graph (or a graph family) is studied, we care about induced rather than partial subgraphs [16].) We currently deal only with graphlets with up to five nodes. Due to the small-world nature of many real-world networks [138], and since the number of graphlets on n nodes increases exponentially with n, we believe that using larger graphlets would unnecessarily increase the computational complexity. There are 30 such graphlets, denoted by $G_0,\ldots,G_{29}$ in Fig. 2.2b. We may further refine the graphlet idea by noticing that in some graphlets, the nodes are topologically distinct from each other. For example, in a ring (cycle) of four nodes (graphlet G_5 in Fig. 2.2b), every node looks the same as every other, but in a chain (path) of four nodes (graphlet G_3 in Fig. 2.2b), there are two end nodes and two middle nodes. These "symmetry groups" within graphlets can be mathematically formalized by using the notion of graph "automorphism orbits" [99]: for example, the middle node in G_1 is topologically distinct from the end nodes of G_1. There are 73 topologically distinct orbits across all 30 2-, 3-, 4-, and 5-node graphlets, labeled from 0 to 72 in Fig. 2.2b. In this way, we greatly enhance the topological sensitivity of using graphlets to characterize network structure and compare networks, without increasing the computational cost.

Graphlet-based systematic measures of local network structure have been proposed that impose a large number of similarity constraints on networks being compared [99, 100]. One approach compares the frequencies of appearance of graphlets in two networks and provides a statistical characterization of their local structural similarity independent of any random network model [100]. Another approach generalizes the degree distribution, which measures the number of nodes in a network having degree k, that is, "touching" k edges (an edge being the only 2-node graphlet), into the spectrum of 73 "graphlet degree distributions," each of which measures the number of nodes in a network touching k orbits of a given type. Comparing 73 graphlet degree distributions of two networks gives a highly constraining measure of agreement between their network topologies [99]. Similar graphlet-based generalization can be applied to the clustering spectrum, where each of the 73 graphlet clustering spectra would measure the average clustering coefficient of all nodes in a network touching k orbits of a given type. Network analysis and modeling software package called GraphCrunch[5] implements graphlet-based approaches for network comparison [81].

When comparing two networks, one of their network properties can tell us that the networks are similar, while another can tell us that they are different.

[5] http://www.ics.uci.edu/~bio-nets/graphcrunch/.

For example, networks G and H in Fig. 2.2a are identical with respect to their degree distributions (as well as with respect to their clustering spectra), but they are different with respect to their graphlet frequencies. There exist approaches that try to overcome such contradictions in the agreement of different network properties. They do so by integrating the variety of global and local properties of a network into the "network fingerprint" and feeding this fingerprint into different machine learning classifiers to classify the network into a certain graph family (see [80] for details).

2.3 Node Centralities

Another important concept is that of node centrality. Node centrality measures try to determine the relative topological importance of a node based on its position in the network. Several centrality measures have been proposed. Examples include the following: (1) *degree centrality*, according to which nodes with high degree have high centrality; (2) *closeness centrality*, according to which nodes with short paths to all other nodes have high centrality; (3) *betweenness centrality*, according to which nodes that occur in many of the shortest paths have high centrality; (4) *eigenvector centrality*, according to which nodes have high centrality if their neighbors also have high centrality (and vice versa); (5) *subgraph centrality*, defined as the sum of closed walks of different lengths in the network starting and ending at the node in question, with smaller lengths having higher importance, where each closed walk is associated with a connected subgraph (closed walks of order n represent subgraphs on n nodes). Hence, this measure counts the number of (partial) subgraphs that a node participates in [25] and according to it, nodes that participate in a large number of subgraphs have high centrality; and (6) *graphlet degree centrality*, that generalizes the degree of a node, which counts the number of edges (i.e., orbits 0) that the node touches, into the *graphlet degree vector* (GDV) of the node, that counts how many of each of the 73 orbits for 2–5-node graphlets the node touches (Fig. 2.2c). The *GDV* of a node thus has 73 elements and it describes the topology of the node's up to 4-deep neighborhood. A measure of similarity between GDVs of two nodes, *GDV-similarity* is defined that quantifies the topological similarity of extended neighborhoods of the two nodes [83] (Fig. 2.2d). Given the GDV of a node, we can define a new centrality measure, *graphlet degree centrality*, according to which nodes that touch many graphlets, that is, their orbits, have high centrality. It represents the sum of the values of 73 coordinates of the GDV, weighted to account for orbit "dependencies" (see [83] for details).

In Sects. 3–5, we discuss biological applications of different network properties and node centrality measures, including identification of a good network model for biological networks, design of an effective cost function for topological network alignment, protein function prediction, and disease gene identification.

3 Network Modeling

Biological networks are large and complex. To understand them, we must be able to successfully reproduce them. This requires finding a good network model that generates networks that closely replicate the structure of real-world networks. Such a well-fitting model that precisely reproduces the network structure and laws through which the network has emerged can help us understand and replicate the underlying complex evolutionary mechanisms in the cell. Only with the help of such laws, we can make predictions about the phenomenon at study that we may want to further explore experimentally. For example, properties of a model can be used for time- and cost-optimal interactome detection [73] and data denoising [70].

Finding models for biological networks is nontrivial, owing not only to incompleteness and noisiness of the data, but also to computational intractability of many graph theoretic problems: identification of a good network model requires comparing the structure of real-world networks with the structure of model networks, and network properties (described above) must be used to evaluate the fit of a model to the data. Note that, it is the computationally hardness of many graph theoretic problems that gives an additional motivation for finding a well-fitting model for biological networks. Special graph classes often have well-known properties and solving many problems on such classes is feasible even though it is infeasible for graphs in general. Thus, finding an appropriate graph class (i.e., network model) for biological networks could simplify their computational manipulation and enable easier extraction of biological knowledge that is encoded in their network topology.

First, we describe the most commonly used network models (Sect. 3.1). Then, we discuss how they can be used to learn new biology (Sect. 3.2).

3.1 Network Models

Various network (or random graph) models have been proposed for real-world networks that have progressed through a series of versions designed to match certain properties of real-world networks.

The earliest such model is *Erdös–Rényi random graph* model [24]. Erdös–Rényi random graphs are based on the principle that the probability that there exists an edge between any pair of nodes is distributed uniformly at random (Fig. 2.3a). Erdös and Rényi have defined several variants of the model. The most commonly studied one is denoted by $G_{n,p}$, where each possible edge in the graph on n nodes is present with probability p and absent with probability $1 - p$. Many of the properties of Erdös–Rényi random graphs are mathematically well understood [13]. Therefore, it is a standard model to compare the data against, even though it is not expected to fit the data well. Erdös–Rényi graphs have small diameters, “bell-shaped” Poisson degree distributions, and low clustering coefficients, and thus do not provide a good

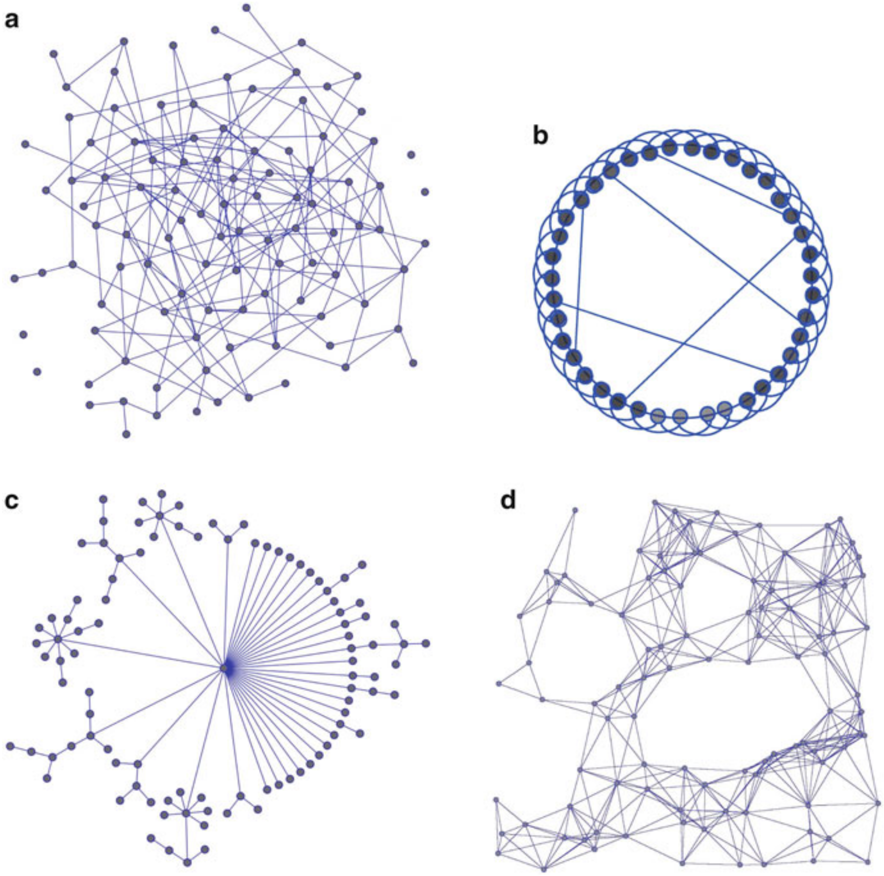


Fig. 2.3 Examples of model networks. (**a**) An Erdös–Rényi random graph. (**b**) A small-world network. (**c**) A scale-free network. (**d**) A geometric random graph

fit to real-world PPI networks which typically have small diameters, but power-law degree distributions and high clustering coefficients. Hence, other network models for real-world networks have been sought.

One such model is *generalized random graph* model. In these graphs, edges are randomly chosen as in Erdös–Rényi random graphs, but the degree distribution is constrained to match the degree distribution of the data [1, 87, 88, 91]. They can be generated by using the "stubs method" [90]: the number of "stubs" (to be filled by edges) is assigned to each node in the model network according to the degree distribution of the real-world network; edges are created between pairs of nodes picked at random; after an edge is created, the number of "stubs" left available at the corresponding "end-nodes" of the edge is decreased by one. Thus, these networks preserve the degree distribution and small diameters of PPI networks. However, they fail to mimic well high clustering coefficients of the data.

Another commonly used network model is that of *small-world* networks. These networks are created from regular ring lattices by random rewiring of a small percentage of their edges (Fig. 2.3b); in a regular ring lattice, nodes are placed on a ring and connected to their *i*th neighbors on the ring for all *i* smaller than some given number *k*. Designed in such a way, small-world networks have large clustering coefficients and diameters that are an order of magnitude smaller than the number of their nodes [92, 93, 138]. However, these networks still fail to reproduce power-law degree distributions of real-world networks.

Scale-free networks are characterized by power-law degree distributions [7, 8, 15, 74, 118] (Fig. 2.3c). Hence, many variants of scale-free network growth models have been proposed for PPI networks. One such model is the Barabási–Albert preferential attachment model [7], in which newly added nodes preferentially attach to existing nodes with probability proportional to the degree of the target node. Other variants focused on modeling PPI networks are based on biologically motivated gene *duplication and mutation* network growth principles [47, 96, 131, 137]: networks grow by duplication of nodes (genes), and as a node gets duplicated, the child node not only inherits most of the interactions of the parent node, but also gains some new interactions. Although scale-free networks have power-law degree distributions and small average diameters, they typically still have low clustering coefficients, and hence they fail to mimic high clustering coefficients of real-world networks.

High clustering coefficients of real-world networks are well reproduced by *geometric graphs* (Fig. 2.3d), which are defined as follows. Nodes in a geometric graph correspond to points distributed in a metric space and edges are created between pairs of nodes if the corresponding points are close enough in the metric space according to some distance norm [97]. For example, 3-dimensional Euclidean boxes and the Euclidean distance norm have been used as a proof of concept to model PPI networks [99, 100]. If the points are distributed in a metric space uniformly at random, then this is a *geometric random graph*. Although this model creates networks with high clustering coefficients and small diameters, it still fails to reproduce power-law degree distributions of real-world PPI networks. Instead, geometric random graphs have Poisson degree distribution. However, it has been argued that power-law degree distributions in PPI networks are an artifact of noise present in them [49, 124]. Moreover, since geometric graphs seem to provide the best fit to the currently available PPI networks [51, 68, 99, 100] and since genomes have evolved through gene duplication and mutation events rather than at random, new models that bridge the concepts of network geometricity with the evolutionary dynamics have been introduced [103]. These *geometric gene mutation and duplication* models are based on the following observations. All biological entities, including genes and proteins as gene products, exist in some multidimensional biochemical space. Genomes evolve through a series of gene duplication and mutation events, which are naturally modeled in the above mentioned biochemical space: when a gene gets duplicated, the child starts at the same point in biochemical space as its parent, and then natural selection acts either to eliminate one, or causes them to separate in the biochemical space (via mutations). This means that the child inherits some of the neighbors of its parent while possibly gaining novel connections as well. The further the child is moved away from its parent, the more different their biochemical properties. Consequently, the further the child is moved away from the parent, the smaller the number of their common interacting partners. These processes can naturally be modeled by geometric graphs. Although motivated by biological principles, these current geometric network models are quite crude mathematical approximations of real biology and further refinements are necessary for obtaining well fitting models for PPI networks.

Finally, biologically motivated *stickiness network* model is based on stickiness indices, numbers that summarize node connectivities and thus also the complexities of binding domains of proteins in PPI networks. The probability that there is an edge between two nodes in a "sticky" model network is directly proportional to the stickiness indices of nodes, that is, to the degrees of their corresponding proteins in real-world PPI networks (see [102] for details). Networks produced by this model have the expected degree distribution of a real-world network. Additionally, they mimic well the clustering coefficients and the diameters of real-world networks.

Illustrations of networks of about the same size that belong to different network models are presented in Fig. 2.3; even without computing any network property for them, we can conclude that their structure is very different just by looking at them. Network analysis and modeling software package called GraphCrunch[6] is capable of evaluating the fit of a series of network models to the data with respect to a variety of global and local network properties [81].

Early studies that published largely incomplete Y2H PPI data sets and that were based on the assumption that the degree distribution was one of the most important network properties that a good network model should capture, tried to model the data with scale-free networks. However, networks of vastly different structure can have the same degree distribution (Fig. 2.2a), and hence it might not be an appropriate measure of network structural similarity. As new biological network data becomes available, we need to ensure that our models continue to fit the data well. In the light of new PPI network data, several studies have started questioning the wellness of the fit of scale-free models. New, more constraining graphlet-based measures of local network structural similarity suggested that the structure of newer and more complete PPI networks is closer to geometric graphs [99, 100]. The geometricity of PPI networks has additionally been supported by demonstrating that PPI networks can explicitly be embedded into a low-dimensional geometric space [51]. The geometric graph model has further been refined to fit the data by learning the distribution of proteins in that space [69]. Moreover, biological reasons why PPI networks are geometric have been argued [103] (see above for the description of geometric gene duplication and mutation models).

Note that different network models can be identified as the best-fitting to the data with respect to different network properties. This raises the issue of which property to trust when evaluating the fit of a model to the data. In general, local network properties are more constraining measures of network structural similarity

[6] http://www.ics.uci.edu/~bio-nets/graphcrunch/.

than global ones (Sect. 2). Also, there exist approaches that try to find a consensus between models suggested by different network properties [80]. They do so by integrating a series of global and local network properties to evaluate the fit of different models to the data using several machine learning classifiers, thus increasing our confidence in the identified best-fitting model [80]. This integrative approach has confirmed that structure of PPI networks is the most consistent with the structure of geometric graphs [80].

3.2 Biological Applications

Despite the low quality of currently available PPI network data and the crudeness and primitivity of existing network models, the models have already been used in practical biological applications to address realistic problems.

As mentioned above, network models are crucial for discovery of network motifs, which are believed to correspond to evolutionary conserved functional building blocks of biological networks; recall that network motifs are defined with respect to a random graph model [86, 117]. Similarly, network models are essential when motifs are used to classify real-world networks into super-families [85]. Furthermore, network models can be used as cost-effective strategies for completing interaction maps, which is an active research topic (e.g., see [110]). A scale-free network model was used in 2004 to guide biological experiments for data collection in a time- and cost-optimal way, thus minimizing the costs of interactome detection [73]. Recall that scale-free networks contain hubs. With this in mind, pull-down experiments were designed to perform "optimal walk" through the PPI network, so that hubs were preferentially chosen as baits. This strategy would allow for time- and cost-optimal detection of most of the interactions with the minimum number of expensive pull-down experiments; however, there is a danger of using inadequate network models for such a purpose. At best, this would result in unsuccessful discovery of the interactome and thus in wasted time and resources. At worst, this could result in wrong identification of "complete" interactome maps, since inadequate models might prevent us from ever examining certain parts of the interactome. In addition to interactome detection, properties of a network model were used to develop computationally easy algorithms for PPI networks that are computationally intensive on graphs in general [101]. Specifically, geometric graph model has been used for designing efficient algorithms for graphlet count estimation [101]. Another application of geometric graph model is denoising of PPI network data: this model has been used to assign confidence levels to existing interactions, as well as to predict new interactions that were overlooked experimentally [70]. For these reasons, and the given above discussion about the scale-freeness of early, incomplete PPI data sets and the geometricity of newer, more complete PPI data sets, it is important to use as accurate models as possible, as well as to ensure that the models keep to fit the data well as the data becomes more complete.

Since discovering PPI and other biological networks is in its infancy, it is expected that practical application of network models will increase and prove its value in the future. The ultimate hope is that a good network model could provide insights into understanding of biological function, disease, and evolution.

4 Network Alignment

Thus far, we discussed network comparison in the context of finding overall similarity between the structure of two networks with respect to a given network property. Another way to think about network comparison is to contrast two networks by finding a mapping between their nodes, with the goal of "fitting" one network into the other as well as possible and identifying topologically similar and functionally conserved network regions. This type of network comparison is referred to as *network alignment* (Fig. 2.4). In addition, there exist some other types of network comparison, such as network integration and network querying (for details, see [114]). The focus of this section is on network alignment. We briefly discuss data (network) integration in Sect. 6.

Network alignment is expected to have at least as deep and valuable impact on our understanding of evolution, biology, and disease as genomic sequence alignment has had. The ability to compare biological networks of different species could enable transfer of knowledge between species, since we may know a lot about some nodes in one network and almost nothing about topologically similar nodes in the other network. This could provide insights into, for example, conservation of protein function or PPIs across species. Moreover, network alignments can be used to measure the global similarity between biological networks of different species and the resulting global network similarities can be used to infer phylogenetic relationships, with the intuition that species with more similar network topologies should be closer in the phylogenetic tree [68, 84]. Meaningful biological network comparisons can be seen as one of the most prominent problems in evolutionary and systems biology.

Unfortunately, unlike with the sequence alignment, the problem of network alignment is computationally infeasible to solve exactly, owing to the NP-completeness of the underlying subgraph isomorphism problem. As mentioned above, even if the subgraph isomorphism problem was feasible, it is unlikely that one biological network would exist as an exact subgraph of another due to noise in the data [133] and also due to biological variation. Hence, approximate algorithms for network alignment need to be sought. In addition, since network alignment requires "fitting" one network into another even if it does not exist as an exact subgraph of the other one [114], it is not obvious how to measure the "goodness" of this fit, and heuristic solutions are needed for this purpose as well.

Analogous to sequence alignments, there exist *local* and *global* network alignments. Local network alignments map independently each local region of similarity, thus resulting in ambiguous one-to-many node mappings (Fig. 2.4a). On the other

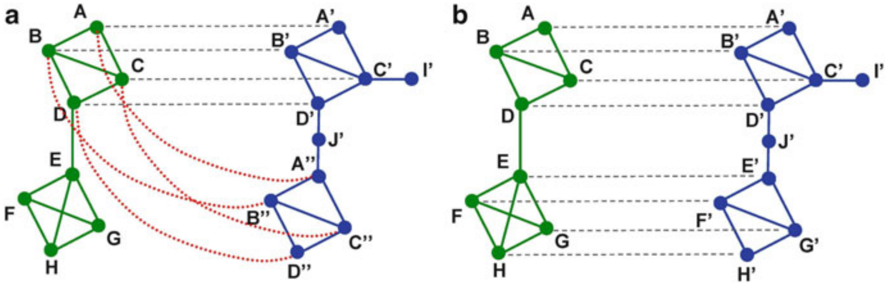


Fig. 2.4 An illustration of alignment of two networks: (**a**) local network alignment and (**b**) global network alignment. In panel (**a**), nodes A, B, C, and D in the network on the *left* (*green*) are aligned to nodes A', B', C', and D', respectively, in the network on the *right* (*blue*), as indicated by *dashed grey lines*. At the same time, these nodes are aligned to nodes A", B", C", and D", respectively, in the network on the *right* (*blue*), as indicated by *dotted red lines*. Thus, local network alignment allows for one-to-many node mappings. Moreover, some of the nodes from the network on the *left* (*green*) remained unaligned. In panel (**b**), with global network alignment, each node in the network on the *left* (*green*) is aligned to a single (unique) node in the network on the *right* (*blue*), as indicated by *dashed grey lines*

hand, global network alignments map uniquely each node in the smaller network to only one node in the larger network, even though this may lead to suboptimal matchings in some local regions (Fig. 2.4b). The majority of currently available algorithms for aligning biological networks have focused on local alignments [5, 9, 10, 40, 76].

4.1 Local Network Alignments

Local network alignments aim to identify small subnetworks that are believed to represent biological pathways or protein complexes that have been evolutionary conserved in PPI networks of different species [5, 9, 10, 40, 76]. The earliest such algorithm is PathBLAST that searches for high-scoring alignments of pathways between two PPI networks. It does so by taking into account the probabilities that PPIs in a pathway are true PPIs rather than false-positives, as well as the homology information derived from sequences of the aligned proteins, while allowing for "gaps" (interacting proteins in one pathway being aligned with sequence-similar proteins in the other pathway that do not interact directly) and "mismatches" (aligned proteins not being sequence-similar) [5]. PathBLAST identified orthologous pathways between baker's yeast and bacterium *H. pylori*. It also detected substantial differences between PPI network of *P. falciparum* and PPI networks of other eukaryotes [125]. PathBLAST was later extended into NetworkBLAST-M to allow for identification of conserved protein complexes rather than pathways in multiple species [116].

Another approach for identifying conserved clusters, called Maximum Weight Induced Subgraph (MaWISh), constructs a weighted global alignment graph and tries to identify a maximum weight induced subgraph in it. It extends the concepts of evolutionary events in sequence alignments to that of duplication, match, and mismatch in network alignments, and it evaluates the similarity between network structures through a scoring function that accounts for these evolutionary events [66]. MaWISh was used to perform pairwise alignments of baker's yeast, worm *C. elegans*, and fruitfly *D. melanogaster* PPI networks. Graemlin, another local network aligner, enables searching for dense networks of an *arbitrary* structure and aligning multiple networks. It is a greedy seed-and-extend approach that gives a score to a possibly conserved module by computing the log-ratio of the probability that the module is subject to evolutionary constraints and the probability that the module is under no constraints, while taking into account phylogenetic relationships between species whose networks are being aligned [40]. It was applied to ten microbial PPI networks and it successfully aligned known biologically functional modules.

Since local network alignments are generally not able to identify large subgraphs that have been conserved during evolution [5, 9], algorithms for global network alignment have also been proposed [26, 41, 68, 84, 120, 121, 145].

4.2 Global Network Alignments

The earliest algorithm for global network alignment, IsoRank, uses spectral graph methods and formulates an eigenvector problem to compute topological scores for aligning nodes from different networks. It uses the intuition that two nodes (one node from each network) should be topologically matched only if their neighbors can also be topologically matched [120]. Next, IsoRank combines the resulting topological node alignment scores with BLAST scores [3] quantifying proteins' sequence similarities. These combined node alignment scores are then used to construct an alignment by the repetitive greedy strategy of identifying and outputting the highest scoring pair and removing all scores involving any of the two identified nodes. Hence, IsoRank combines topology with sequence information to construct an alignment. IsoRank was able to identify functional orthologs between baker's yeast and fruitfly. Other algorithms focusing on aligning baker's yeast and fruitfly have been proposed that outperform IsoRank [145]. IsoRank was later extended to perform local and global alignments of multiple networks [26, 121]. The most recent version, IsoRankN, relies on the notion of node-specific rankings and uses a method similar to PageRank-Nibble algorithm [26]. Although it is a global alignment algorithm in the sense that each node in the smaller network is aligned to a node in the larger network, IsoRankN's alignment contains sets of aligned nodes, where no two sets overlap, but each set can contain more than one node from each of the networks being aligned, thus allowing for many-to-many node mappings. IsoRankN was used to align PPI networks of five eukaryotic species: baker's yeast, fruitfly, worm, mouse, and human. Graemlin has also been extended to allow for global network alignments. It uses a learning algorithm that takes as input a training set of known network alignments, as well as phylogenetic relationships between species whose networks are being aligned, to learn parameters for its scoring function. Then, it automatically adapts the learned objective function to any set of networks. Graemlin's scoring function that incorporates seven evolutionary events allows it to align multiple networks in linear time. It was used to align up to six PPI networks of both eukaryotic and prokaryotic species.

4.2.1 Topological Network Alignments

Most of the existing network alignment methods incorporate some biological information external to network topology. For example, they use some a priori information about nodes, such as sequence similarities of proteins in PPI networks [10, 120], or they use some form of learning on a set of "true" alignments [41]. Hence, network alignment approaches have been sought that rely solely and explicitly on network topology [68, 84] (see below). The development of such topology-based methods is motivated by the following. First, sequence alignment algorithms do not use biological information external to sequences to perform alignments and neither should network alignment algorithms use biological information external to network topology. Since sequence and network topology have been shown to provide complementary insights into biological knowledge [79], and since network topology describes an important part of biological information (as do sequences), using sequence information could deter from finding biological information that is encoded in network topology. And we believe that it is scientifically interesting to ask how much biological information could be extracted from topology only. Second, we need to ensure that high biological quality of an alignment (e.g., orthology between aligned proteins or conservation of biological function between aligned subnetworks) is achieved by the alignment algorithm, and not by using biological data sources external to network topology, such as protein sequence information. It has been argued that only after reliable algorithms for purely topological network alignments have been developed that result in alignments of good biological quality, it would be beneficial to integrate them with other sources of biological information to improve their quality [68]. We believe that we would be able to exploit efficiently various sources of biological information only after we have reliable topology-based network alignment algorithms. Hence, network alignment algorithms based on topology only are expected to be of great importance.

GRAph ALigner (GRAAL) [68] and Hungarian algorithm-based GRAAL (H-GRAAL) [84] rely on topology only rather than on both sequence and topology, as do other algorithms. Hence, they can align networks of any type, not only biological ones. This is important, since network alignment has applications across an enormous span of domains. Both GRAAL and H-GRAAL are based on the same node alignment cost function, namely GDV-similarity (Sect. 2.3). They differ as follows. GRAAL is a *greedy* "seed and extend" approach that, analogously to the popular BLAST algorithm for sequence alignment, first chooses a "seed" node pair (one node from each network) with high GDV-similarity and then expands the alignment radially outward around the seed as far as is practical using a greedy algorithm, with the goal of quickly finding *approximate* alignments [68]. On the other hand, H-GRAAL is a more expensive search algorithm that uses Hungarian algorithm for solving the assignment problem [71, 139] to find *optimal* alignments relative to the alignment cost function; the Hungarian algorithm [71] is a combinatorial optimization algorithm for finding a maximum weight matching in a weighted bipartite graph.

When applied to baker's yeast and human PPI networks, GRAAL and H-GRAAL expose regions of network similarity about an order of magnitude larger than other algorithms, indicating that even distant species share a surprising amount of network topology and potentially suggesting broad similarities in internal cellular wiring across all life on Earth. In addition to such high topological quality of their alignments, GRAAL and H-GRAAL also recover the underlying biological function, in the sense that a large number of aligned yeast-human protein pairs and subnetworks are involved in a same biological process. Hence, the two algorithms are used to transfer biological function from annotated to unannotated parts of aligned networks, and many of such functional predictions are validated in the literature. Analogous to sequence alignments, they are also used to infer phylogeny as follows [68, 84]. Metabolic networks of closely related species are aligned with these algorithms and the resulting alignment scores, quantifying the level of similarities between networks of different species, are used to infer species' phylogeny, with the intuition that more similar networks should be closer in the phylogenetic tree. Phylogenetic trees constructed from GRAAL's and H-GRAAL's purely topological alignments for protists and fungi closely resemble those found by sequence comparisons, indicating that network topology and network alignments in general could potentially provide a new, independent source of biological and phylogenetic information.

5 Interplay Between Network Topology and Biological Function and Disease

Since proteins aggregate to perform a function instead of acting in isolation, and since PPI networks model interactions between proteins, analyzing PPI network topology is expected to uncover new biology. Therefore, it is not surprising that numerous approaches attempting to link network topology with biological function have been proposed. Network-based prediction of protein function [113] (Sect. 5.1) and the role of protein networks in disease [115] (Sect. 5.2) have received much attention in the postgenomic era.

5.1 From PPI Networks to Biological Function

The earliest approaches attempted to link degrees of proteins in PPI networks with their biological function. Initial studies observed good correlations between proteins' essentiality and their degree centralities in a PPI network of baker's yeast [57]. However, the controversy arose in the light of newer and more complete PPI network data for which this correlation was not observed [107, 144]. It appears to hold only for literature-curated [38] and smaller in scope Y2H [128] PPI networks, possibly because these data sets are biased toward essential proteins [107].

Considering node degree as a measure of topological positioning of a protein in a PPI network led to another controversy [27, 33–35]. High-degree nodes in a PPI network, called "hubs," have been studied in the context of expression correlation, colocalization, evolutionary rates, and structural perturbation of a PPI network upon deletion. However, the results could not be reproduced on literature-curated PPI network data sets [34, 35]. Reasons for this controversy could be as follows. First, degree alone might be a weak measure of network topology, as it captures limited network topology, that is, only direct neighborhood of a node. Second, a distinction between two types of hubs in a PPI network has been postulated: some of them are "party" hubs whose genes are coexpressed with all of their neighbors' genes over many physiological conditions and are thus concurrently interacting with all of their neighbors, while others are "date" hubs whose genes are coexpressed with only one or few neighbors' genes in each physiological condition and are thus consecutively interacting with their neighbors [27, 33]; the latter are thus not true hubs, since their degrees are low and depend on the physiological state. Hence, the controversy may have resulted from biases that different techniques for PPI network construction impose on PPI network topologies [32, 133]. An example are the matrix and spoke models [141] used to define PPIs in a protein complex pulled down in an AP/MS experiment (Sect. 1 and Fig. 2.2d). The choice of a model has been shown to have a strong impact on the PPI network topology [48].

Similar simple correlations between node degrees in a PPI network and protein function were examined [104]. A variety of approaches have been relying on the assumption that proteins that are closer in a network are more likely to perform the same function [18, 113]. In the most simple form, this assumption has been used to investigate the direct neighborhood of an unannotated protein and annotate it with the most common functions among its annotated neighbors [111] (Fig. 2.5a). Related approaches have analyzed a deeper n-neighborhood of a protein, defined as a set of proteins that are at most at distance n from the protein [52], or assigned weights to proteins at different distances from the protein of interest [18]. The idea of shared network neighbors has been used: proteins that share many common neighbors also have close functional associations [109] (Fig. 2.5a). Additionally, a global optimization-based function prediction strategy has been proposed: any given assignment of functions to the whole set of unclassified proteins in a network is given a score, counting the number of interacting pairs of nodes with no common function; the functional assignment with the lowest score maximizes the presence

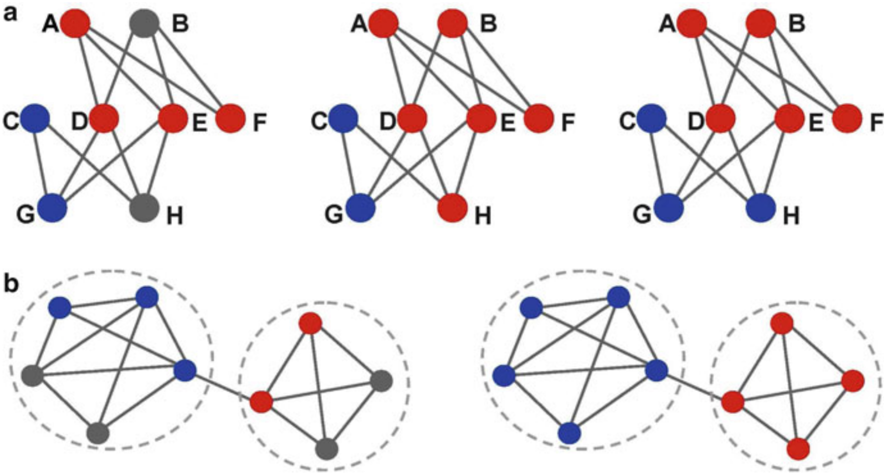


Fig. 2.5 Examples of approaches for protein function prediction. (**a**) In the network on the *left*, nodes A, D, E, and F are annotated with function "red," nodes C and G are annotated with function "blue,", and proteins B and H are unannotated (as denoted by *grey node color*). With the "direct neighborhood" approach (the *middle panel*), node B gets annotated with "red," since all of its neighbors (D, E, and F) have function "red." Similarly, H gets annotated with "red," since the majority of its neighbors (D and E) have function "red." With the "shared neighbors" approach (the *right panel*), node B gets annotated with "red," since it shares the largest number of neighbors with A, which has function "red." However, with this approach, H gets annotated with "blue," since it shares the largest number of neighbors with G, which has function "blue." (**b**) With cluster-based approaches for protein function prediction, the network is first partitioned into clusters, denoted by *dashed grey circles* (the *left panel*) and then each cluster gets a function based on the function of its annotated members (the *right panel*)

of the same function among interacting proteins [132]. A network flow-based idea has also been suggested: each functionally annotated protein in the network is considered as the source of a "functional flow" and the spread of the functional flow through the network is simulated over time; then, each unannotated protein is assigned a score for having the function based on the amount of flow it received during the simulation [89]. Also, functional homogeneity of groups of proteins that show some type of "coherence" in the PPI network, called clusters, has been used for protein function prediction [6,65,67,104,112] (Fig. 2.5b): the idea is to partition the network into clusters and assign the entire cluster with a function based on the functions of its annotated members. Due to the above mentioned controversies linked to using overly simple measures of network topology, such as node degrees, more recent studies have been relying on GDVs, a highly constraining measure of network topology that is based on all 2–5-node graphlets and that captures up to 4-deep network neighborhood of a node, compared to degree alone that is based on the only 2-node graphlet and that captures only 1-deep neighborhood (Sect. 2.3). The correlation between proteins' GDV-similarities in a PPI network, that is, similarities of their extended network neighborhoods, and the similarity of their biological function has been observed and used to predict function of unannotated proteins [26,83].

5.2 PPI Networks in Disease and Pharmacology

Characterizing and identifying novel disease genes represents an opportunity for therapeutic intervention, since these genes represent potential drug targets and could thus lead to better drug design. An assessment of the number of drug targets, as well as their identification, is crucial for the development of postgenomic research strategies within the pharmaceutical industry [54]. Now that the size of the human genome is known, it is interesting to consider just how many molecular targets this opportunity represents. Out of all types of macromolecules with which small-molecule therapeutic agents can interfere, the majority of successful drugs achieve their activity by binding to, and modifying the activity of, a protein. The human PPI network can assist in investigating network properties of disease genes and identifying novel disease genes. Also, network-based analyses can help understand the relationships between genetic disorders and genes causing the disorders, as well as between drugs and their protein targets [115].

Inspired by the findings that essential yeast proteins tend to have high degrees in PPI networks, several studies attempted to perform similar analyses on disease-related genes. They examined whether disease-related genes (i.e., proteins as gene products) could be distinguished based on their topological properties and position in the PPI network. When only node degrees were used to measure topology, a discrepancy was observed again in the sense that some groups reported that genes (proteins) involved in disease tend to have high degrees in PPI networks [59, 136], while others contradicted that conclusion by arguing that the observed correlation between high degrees and disease genes was entirely due to the existence of essential genes within the disease gene class [28]. Apart from this, general conclusions are that disease genes have high connectivity, are closer together, and are centrally positioned within the PPI network [115]. However, these results might be biased, since disease proteins may exhibit these properties in a PPI network simply because they have been better studied than nondisease proteins.

Given these central topological roles of disease-related proteins in the human PPI network, a straightforward step was to identify candidate disease genes from network topology. The key assumption of most such approaches is that a neighbor of a disease-causing gene in a PPI network is likely to cause either the same or a similar disease [115]. For example, Aragues et al. [4] started from the hypothesis that proteins whose partners have been annotated as cancer genes are likely to be cancer genes as well, constructed a cancer protein interaction network composed of known cancer genes and their direct interacting partners, and demonstrated that the "cancer linker degree" of a protein, that is, the number of its cancer-related neighbors in this network, is a good indicator of the probability that the gene is a cancer gene. Radivojac et al. [105] have tried to identify gene-disease associations by encoding each gene in a PPI network based on the distribution of shortest path lengths to all genes associated with disease or having known functional annotation. Similarly, propagating the "flow" from disease-causing proteins to their neighbors in the PPI network was used to score the strength of association of proteins and protein complexes with a disease [130]. Graphlets have also been used to relate proteins' network topological characteristics to their involvement in disease [82, 83]: GDV-similar proteins, that is, proteins with topologically similar network neighborhoods, were grouped together and the resulting clusters were found to be statistically significantly enriched in known cancer genes; as such, they were used to predict novel cancer gene candidates [82]. In addition, by observing only the topology around nodes in PPI networks and finding nodes that are topologically similar to nodes that are known regulators of melanogenesis, novel regulators of melanogenesis were identified in human cells and the predictions were phenotypically validated by systems-level functional genomics siRNA screens [42, 82].

Moreover, PPI networks have been combined with the networks describing the relationships between diseases and genes causing them [28]. The "diseasome," the combined set of all known associations between human genetic disorders and the respective disease genes, was used to create: (a) a network in which nodes are genetic disorders and two nodes are connected if the same disease gene was implicated in both disorders; and (b) a network in which nodes are disease genes and two nodes are connected if they are both implicated in the same disease. By analyzing the former, it was shown that the genetic origins of most diseases were shared with other diseases to some extent. By analyzing the latter and overlaying it with the PPI network, genes that contributed to a common disorder showed an increased tendency for their protein products to interact through PPIs, be expressed together in specific tissues, display high coexpression levels, exhibit synchronized expression as a group, and perform same biological function [28].

Similarly, PPI networks have recently been combined with the networks describing the relationships between drugs and their protein targets [31]. Such network-based analyses of drug action are starting to be used as part of an emerging field of *systems pharmacology*, which aims to understand drug action across multiple scales of complexity, from cellular to tissue to organismal [11]. Multiple studies have constructed network types that link biochemical interaction networks, such as PPI networks, with networks of drug similarities or therapeutic indications. For example, similar to the above "diseasome," the combined set of all known associations between drugs and their protein targets was used to generate: (a) a network in which nodes are drugs and they are connected if they share a common target; and (b) a network in which nodes are targets and they are connected if they are affected by a same drug [31]. By analyzing the former and by taking into consideration the time the drug was introduced, it was shown that there are relatively few drugs acting on novel targets that enter the market. By analyzing the later and overlaying it with the PPI network, drug targets showed the tendency to have higher degrees than nontargets in the PPI network. However, as mentioned above, this observation might be an artifact of disease-related parts of the PPI network receiving more attention. For a survey of network-based analyses in systems pharmacology, see [11]. The drug and drug-target data can be found in DrugBank [140], a comprehensive dual purpose bioinformatics–cheminformatics database that brings together chemical, physical, pharmaceutical, and biological data about thousands of well studied drugs and drug targets.

6 Future Prospects and Concluding Remarks

We have presented computational, graph-theoretic methods for analyzing, modeling, and comparing biological networks, as well as their biological applications that have already given valuable insights into biological function, disease, and evolution. Biological network research is young, and many advances are still to come; all of the major advances that occurred for genetic sequence research can be envisioned for biological network research as well. We believe that we have barely touched the tip of the iceberg, as the field is still in its infancy, rich in many important but yet unsolved problems.

A challenging open problem is that of network integration. Networks of different types are becoming available, such as protein–protein, genetic, and protein–DNA interaction networks, all of which cover different slices of biological information. Integrating them would contribute to a comprehensive view of a cellular system. It still remains unclear how to combine these different data types in a systematic and biologically meaningful manner, and the development of efficient network integration methods is necessary. Some pioneering approaches have been proposed that combine networks of different interaction types defined on the same sets of nodes, with the goal of identifying functional modules supported by multiple interaction types. Most of them combine PPI with genetic interaction (GI) networks; in GI networks, nodes correspond to genes and edges exist between two genes if simultaneous mutations of the two genes cause change in cellular phenotype (lethal or sick), while mutation of each individual gene ("null mutation") results in no phenotypic change [14]. PPI networks were integrated with the information about viable, lethal, and GI mutants and it was suggested that alternate paths exist that bypass viable nodes in PPI networks, which offered an explanation why null mutations of these proteins are not lethal [104]. This was further supported by an observation that GIs tend to bridge genes from redundant pathways rather than within a single pathway [64]. Other related studies applied machine learning methods to graph-theoretic properties of proteins in PPI networks to predict GIs [94]. Also, dense clusters of interactions supported by several network types were found more likely to correspond to protein complexes than dense clusters supported by any one network [29]. Finally, integration of networks encompassing different interaction types, as well as identification of "composite" network motifs arising from such combined network data, was used for protein function and interaction prediction [30, 36, 37, 78, 143]. For a review on methods for integration of PPI and GI networks, see [12].

The field of biological network research is expected to bloom as larger amounts of high-quality biological network data and more sophisticated methods for their analysis become available. Also, to further advance this interdisciplinary field, strong synergy must be formed between biological and computational scientists. Thus far, this link has been relatively weak, which resulted in the use of quite simple computational and mathematical methods for analyzing complex biological network data, even though much stronger mathematical tools have been available.

One example is the preferred use of simple measures of network topology, such as node degrees, which often resulted in inconclusive or controversial results. Such practices of using simple mathematical techniques and developing repulsion toward new "complicated" methods and models could potentially lead to the emergence of overly simplistic doctrines that could slow down the growth of the field and even be misleading.

One example of such doctrines is the scale-free-centric view of biological networks. Wide research community has been applying simple computational approaches, such as degree distributions, to early and noisy PPI networks, thus finding a commonly accepted belief about scale-freeness of complex biological networks [62]. However, the scale-free model has since been shown to be far too simplistic a model for biological networks, clearly demonstrating the need to develop and apply better algorithmic and mathematical tools. Another example is the genome-centric view of biological systems. Genetic sequence research has undoubtedly revolutionized our biological understanding. Nonetheless, our biological understanding is still incomplete and alternative scientific avenues might need to be taken to complement the knowledge learned from sequence. Despite low quality of current PPI networks and crudeness of mathematical methods for their analyses, a compelling evidence has been presented that network topology represents such a complementary source of biological information, as it can uncover new knowledge that often cannot be uncovered by genetic sequence alone. Nonetheless, some members of the research community have kept questioning the value of the network data and challenging the relationship between network topology and biological function, even if supported by biological validation. This has reached such worrisome proportions that a question has been raised whether the community "should keep analyzing PPI network data" at all. In addition, the sequence and network communities might also be playing an unfair game. On one hand, the results demonstrating that networks uncover biology complementary to that uncovered from sequence can be interpreted as being wrong and thus get rejected, since they are not in agreement with the widely accepted sequence-based beliefs. On the other hand, the network-based results that are in agreement with sequence-based ones are often regarded as useless, since they only confirm what sequences tell us. Hence, network-based analyses cannot win in such an unfairly played game. The spread of such scientifically unjustified opinions could negatively affect the availability of financial resources necessary for completing the interactome maps and developing reliable and sophisticated computational tools for solving computationally challenging but cutting edge scientific problems, such as network alignment and network data integration.

Nonetheless, we believe that the research community cannot ignore the valuable biological insights that have already been learned from biological networks and that it is starting to realize the full potential of biological networks. Hence, we are confident that the field of biological network research is likely to stay at the forefront of scientific research in the years to come.

References

1. W. Aiello, F. Chung, and L. Lu. A random graph model for power law graphs. *Experimental Mathematics*, 10:53–66, 2001.
2. U. Alon. Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 8:450–461, 2007.
3. S. F. Altschul, W. Gish, W. Miller, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.
4. R. Aragues, C. Sander, and B. Oliva. Predicting cancer involvement of genes from heterogeneous data. *BMC Bioinformatics*, 9:172, 2008.
5. Y. Artzy-Randrup, S. J. Fleishman, N. Ben-Tal, and L. Stone. Comment on Network motifs: Simple building blocks of complex networks and Superfamilies of evolved and designed networks. *Science*, 305:1107c, 2004.
6. G. D. Bader and C. W. V. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4:2, 2003.
7. A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
8. A.-L. Barabási, R. Albert, and H. Jeong. Mean-field theory for scale-free random networks. *Physica A*, 272:173–197, 1999.
9. J. Berg and M. Lassig. Local graph alignment and motif search in biological networks. *PNAS*, 101:14689–14694, 2004.
10. J. Berg and M. Lassig. Cross-species analysis of biological networks by Bayesian alignment. *Proceedings of the National Academy of Sciences*, 103(29):10967–10972, 2006.
11. S. I. Berger and R. Iyengar. Network analyses in systems pharmacology. *Bioinformatics*, 25:2466–2472, 2009.
12. A. Beyer, S. Bandyopadhyay, and T. Ideker. Integrating physical and genetic maps: from genomes to interaction networks. *Nature Reviews Genetics*, 8:699–710, 2007.
13. B. Bollobas. *Random Graphs*. Academic, London, 1985.
14. C. Boone, H. Bussey, and B. J. Andrews. Exploring genetic interactions and networks with yeast. *Nature Reviews Genetics*, 8:437–449, 2007.
15. S. Bornholdt and H. Ebel. World-wide web scaling exponent from Simon's 1955 model. *Physical Review E*, 64:046401, 2001.
16. A. Brandstadt, L. Van Bang, and J. P. Spinrad. *Graph classes: a survey*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, PA 19104-2688, 1999.
17. A Chatr-aryamontri, A Ceol, D Peluso, A Nardozza, S Panni, F Sacco, M Tinti, A Smolyar, L Castagnoli, M Vidal, ME Cusick, and G Cesareni. Virusmint: a viral protein interaction database. *Nucleic Acids Res*, 37:D669–D673, 2009.
18. HN Chua, WK Sung, and L Wong. Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions. *Bioinformatics*, 22:1623–1630, 2006.
19. S.R. Collins, P. Kemmeren, X.-C. Zhao, J.F. Greenblatt, F. Spencer, F.C.P. Holstege, J.S. Weissman, and N.J. Krogan. Toward a comprehensive atlas of the phyisical interactome of saccharomyces cerevisiae. *Mol. Cell Proteomics*, 6(3):439–450, 2007.
20. S.R. Collins, M. Schuldiner, N.J. Krogan, and J.S. Weissman. A strategy for extracting and analyzing large-scale quantitative epistatic interaction data. *Genome Biology*, 7:R63, 2006.
21. S. Coulomb, M. Bauer, D. Bernard, and M.-C. Marsolier-Kergoat. Gene essentiality and the topology of protein interaction networks. *Proc. Roy. Soc. B.*, 272:1721–1725, 2005.
22. E. de Silva and M.P.H. Stumpf. Complex networks and simple models in biology. *Roy. Soc. Interface*, 2:419–430, 2005.
23. E. de Silva, T. Thorne, P. Ingram, I. Agrafioti, J. Swire, C. Wiuf, and M.P.H. Stumpf. The effects of incomplete protein interaction data on structural and evolutionary inferences. *BMC Biology*, 4(39):1–13, 2006.
24. P. Erdös and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

25. E. Estrada and J. A. Rodríguez-Velázquez. Subgraph centrality in complex networks. *Phys Rev E Stat Nonlin Soft Matter Phys*, 71(5 Pt 2), 2005.
26. C. Guerrero et al. Characterization of the proteasome interaction network using a qtax-based tag-team strategy and protein interaction network analysis. *PNAS*, 105:13333–13338, 2008.
27. J-D J. Han et al. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 430, 2004.
28. K. I. Goh et al. The human disease network. *PNAS*, 104:8685–8690, 2007.
29. K.C. Gunsalus et al. Predictive models of molecular machines involved in caenorhadbitis elegans early embryogenesis. *Nature*, 436:861–865, 2005.
30. L.V. Zhang et al. Motifs, themes and thematic maps of an integrated saccharomyces cerevisiae interaction network. *J. Biol.*, 4(6), 2005.
31. M. A. Yildirim et al. Drug-target network. *Nature Biotechnology*, 25:1119–1126, 2007.
32. M. E. Cusick et al. Literature-curated protein interaction datasets. *Nature Methods*, 6:39–46, 2009.
33. N. Bertin et al. Confirmation of organized modularity in the yeast interactome. *PLoS Biology*, 5:e153, 2007.
34. N. Bertin et al. Still stratus not altocumulus: Further evidence against the date/party hub distinction. *PLoS Biology*, 5:e154, 2007.
35. N. N. Batada et al. Stratus not altocumulus: a new view of the yeast protein interaction network. *PLoS Biology*, 4:e317, 2006.
36. P. Kammeren et al. Protein interaction verification and functional annotation by integrated analysis of genome-scale data. *Mol. Cell*, 9:1133–1143, 2002.
37. S. L. Wong et al. Combining biological networks to predict genetic interactions. *Proc. Natl. Acad. Sci. USA*, 101:15682–15687, 2004.
38. T. Reguly et al. Comprehensive curation and analysis of global interaction networks in saccharomyces cerevisiae. *Journal of Biology*, 5:11, 2006.
39. S. Fields. High-throughput two-hybrid analysis. the promise and the peril. *FEBS J.*, 272:5391–5399, 2005.
40. J. Flannick, A. Novak, S.S. Balaji, H.M. Harley, and S. Batzglou. Graemlin general and robust alignment of multiple large interaction networks. *Genome Res*, 16(9):1169–1181, 2006.
41. J. Flannick, A. F. Novak, C. B. Do, B. S. Srinivasan, and S. Batzoglou. Automatic parameter learning for multiple network alignment. In *RECOMB*, pages 214–231, 2008.
42. AK Ganesan, H Ho, B Bodemann, S Petersen, J Aruri, S Koshy, Z Richardson, LQ Le, T Krasieva, MG Roth, P Farmer, and MA White. Genome-wide siRNA-based functional genomics of pigmentation identifies novel genes and pathways that impact melanogenesis in human cells. *PLoS Genet*, 4(12):e1000298, 2008.
43. M. R. Garey and D. S. Johnson. *Computers and Intractability–A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.
44. A. C. Gavin, M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J. M. Rick, A. M. Michon, C. M. Cruciat, M. Remor, C. Hofert, M. Schelder, M. Brajenovic, H. Ruffner, A. Merino, K. Klein, M. Hudak, D. Dickson, T. Rudi, V. Gnau, A. Bauch, S. Bastuck, B. Huhse, C. Leutwein, M. A. Heurtier, R. R. Copley, A. Edelmann, E. Querfurth, V. Rybin, G. Drewes, M. Raida, T. Bouwmeester, P. Bork, B. Seraphin, B. Kuster, G. Neubauer, and G. Superti-Furga. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141–147, 2002.
45. AC Gavin, P Aloy, P Grandi, R Krause, M Boesche, M Marzioch, C Rau, LJ Jensen, S Bastuck, B Dumpelfeld, A Edelmann, MA Heurtier, V Hoffman, C Hoefert, K Klein, M Hudak, AM Michon, M Schelder, M Schirle, M Remor, T Rudi, S Hooper, A Bauer, T Bouwmeester, G Casari, G Drewes, G Neubauer, JM Rick, B Kuster, P Bork, RB Russell, and G Superti-Furga. Proteome survey reveals modularity of the yeast cell machinery. *Nature*, 440(7084):631–636, 2006.
46. L Giot, JS Bader, C Brouwer, A Chaudhuri, B Kuang, Y Li, YL Hao, CE Ooi, B Godwin, E Vitols, G Vijayadamodar, P Pochart, H Machineni, M Welsh, Y Kong, B Zerhusen, R Malcolm, Z Varrone, A Collis, M Minto, S. Burgess, L McDaniel, E Stimpson, F Spriggs,

J Williams, K. Neurath, N Ioime, M Agee, E Voss, K Furtak, R Renzulli, N Aanensen, S Carrolla, E Bickelhaupt, Y Lazovatsky, A DaSilva, J Zhong, CA Stanyon, RL Jr Finley, KP White, M Braverman, T Jarvie, S Gold, M Leach, J Knight, RA Shimkets, MP McKenna, J Chant, and JM Rothberg. A protein interaction map of drosophila melanogaster. *Science*, 302(5651):1727–1736, 2003.

47. K.-I. Goh, B. Kahng, and D. Kim. Hybrid network model: the protein and the protein family interaction networks. *arXiv:q-bio.MN/0312009 v2, 28 March 2004*, 2004.
48. L. Hakes, J.W. Pinney, D. L. Robertson, and S. C. Lovell. Protein-protein interaction networks and biology–what's the connection? *Nature Biotechnology*, 26(1):69–72, 2008.
49. J. D. H. Han, D. Dupuy, N. Bertin, M. E. Cusick, and Vidal. M. Effect of sampling on topology predictions of protein-protein interaction networks. *Nature Biotechnology*, 23:839–844, 2005.
50. Christopher T. Harbison, D. Benjamin Gordon, Tong I. Lee, Nicola J. Rinaldi, Kenzie D. Macisaac, Timothy W. Danford, Nancy M. Hannett, Jean-Bosco Tagne, David B. Reynolds, Jane Yoo, Ezra G. Jennings, Julia Zeitlinger, Dmitry K. Pokholok, Manolis Kellis, P. Alex Rolfe, Ken T. Takusagawa, Eric S. Lander, David K. Gifford, Ernest Fraenkel, and Richard A. Young. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104, 2004.
51. D.J. Higham, M. Rašajski, and N. Pržulj. Fitting a geometric graph to a protein-protein interaction network. *Bioinformatics*, 24(8):1093–1099, 2008.
52. H. Hishigaki, K. Nakai, T. Ono, A. Tanigami, and T. Takagi. Assessment of prediction accuracy of protein function from protein-protein interaction data. *Yeast*, 18:523–531, 2001.
53. Y. Ho, A. Gruhler, A. Heilbut, G. D. Bader, L. Moore, S. L. Adams, A. Millar, P. Taylor, K. Bennett, K. Boutilier, L. Yang, C. Wolting, I. Donaldson, S. Schandorff, J. Shewnarane, M. Vo, J. Taggart, M. Goudreault, B. Muskat, C. Alfarano, D. Dewar, Z. Lin, K. Michalickova, A. R. Willems, H. Sassi, P. A. Nielsen, K. J. Rasmussen, J. R. Andersen, L. E. Johansen, L. H. Hansen, H. Jespersen, A. Podtelejnikov, E. Nielsen, J. Crawford, V. Poulsen, B. D. Sorensen, J. Matthiesen, R. C. Hendrickson, F. Gleeson, T. Pawson, M. F. Moran, D. Durocher, M. Mann, C. W. Hogue, D. Figeys, and M. Tyers. Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry. *Nature*, 415(6868):180–183, 2002.
54. A. L. Hopkins and C. R. Groom. The druggable genome. *Nature Reviews Drug Discovery*, 1:727–730, 2002.
55. T. Ito, K. Tashiro, S. Muta, R. Ozawa, T. Chiba, M. Nishizawa, K. Yamamoto, S. Kuhara, and Y. Sakaki. Toward a protein-protein interaction map of the budding yeast: A comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proc Natl Acad Sci U S A*, 97(3):1143–1147, 2000.
56. S. Itzkovitz, R. Levitt, N. Kashtan, R. Milo, M. Itzkovitz, and U. Alon. Coarse graining and self-dissimilarity of complex networks. *Physical Review E*, 71:016127, 2005.
57. H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, 2001.
58. H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.
59. P. F. Jonsson and P. A. Bates. Lobal topological features of cancer proteins in the human interactome. *Bioinformatics*, 22:2291–2297, 2006.
60. S. Kaplan, A. Bren, E. Dekel, and U Alon. The incoherent feed-forward loop can generate non-monotonic input functions for genes. *Molecular Systems Biology*, 4:203, 2008.
61. S. Kaplan, A. Bren, A. Zaslaver, E. Dekel, and U. Alon. Diverse two-dimensional input functions control bacterial sugar genes. *Molecular Cell*, 29:786–792, 2008.
62. E. F. Keller. Revisiting scale-free networks. *BioEssays*, 27:11060–11068, 2005.
63. B. P. Kelley, Y. Bingbing, F. Lewitter, R. Sharan, B. R. Stockwell, and T. Ideker. PathBLAST: a tool for alignment of protein interaction networks. *Nucl. Acids Res.*, 32:83–88, 2004.
64. R. Kelley and T. Ideker. Systematic interpretation of genetic interactions using protein networks. *Nature Biotechnology*, 23:561–566, 2005.

65. A. D. King, Pržulj, N., and I. Jurisica. Protein complex prediction via cost-based clustering. *Bioinformatics*, 20(17):3013–3020, 2004.
66. M. Koyutürk, Y. Kim, U. Topkara, S. Subramaniam, W. Szpankowski, and A. Grama. Pairwise alignment of protein interaction networks, 2006.
67. NJ Krogan, G Cagney, H Yu, G Zhong, X Guo, A Ignatchenko, J Li, S Pu, N Datta, AP Tikuisis, T Punna, JM Peregrn-Alvarez, M Shales, X Zhang, M Davey, MD Robinson, A Paccanaro, JE Bray, A Sheung, B Beattie, DP Richards, V Canadien, A Lalev, F Mena, P Wong, A Starostine, MM Canete, J Vlasblom, S Wu, C Orsi, SR Collins, S Chandran, R Haw, JJ Rilstone, K Gandi, NJ Thompson, G Musso, P St Onge, S Ghanny, MH Lam, G Butland, AM Altaf-Ul, S Kanaya, A Shilatifard, E O'Shea, JS Weissman, CJ Ingles, TR Hughes, J Parkinson, M Gerstein, SJ Wodak, A Emili, and JF Greenblatt. Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. *Nature*, 440:637–643, 2006.
68. O. Kuchaiev, T. Milenkovic, V. Memisevic, W. Hayes, and N. Pržulj. Topological network alignment uncovers biological function and phylogeny. *Journal of the Royal Society Interface*, 2010.
69. O. Kuchaiev and N. Pržulj. Learning the structure of protein-protein interaction networks. *2009 Pacific Symposium on Biocomputing (PSB)*, 2009.
70. O. Kuchaiev, M. Rasajski, D. Higham, and N. Pržulj. Geometric de-noising of protein-protein interaction networks. *PLoS Computational Biology*, 5:e1000454, 2009.
71. Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.
72. Douglas J. LaCount, Marissa Vignali, Rakesh Chettier, Amit Phansalkar, Russell Bell, Jay R. Hesselberth, Lori W. Schoenfeld, Irene Ota, Sudhir Sahasrabudhe, Cornelia Kurschner, Stanley Fields, and Robert E. Hughes. A protein interaction network of the malaria parasite plasmodium falciparum. *Nature*, 438:103–107, 2005.
73. M. Lappe and L. Holm. Unraveling protein interaction networks with near-optimal efficiency. *Nature Biotechnology*, 22(1):98–103, 2004.
74. L. Li, D. Alderson, R. Tanaka, J. C. Doyle, and W. Willinger. Towards a theory of scale-free graphs: definition, properties, and implications (extended version). *arXiv:cond-mat/0501169*, 2005.
75. S Li, CM Armstrong, N Bertin, H Ge, S Milstein, M Boxem, P-O Vidalain, J-DJ Han, A Chesneau, T Hao, N Goldberg, DS Li, M Martinez, J-F Rual, P Lamesch, L Xu, M Tewari, SL Wong, LV Zhang, GF Berriz, L Jacotot, P Vaglio, J Reboul, T Hirozane-Kishikawa, Q Li, HW Gabel, A Elewa, B Baumgartner, DJ Rose, H Yu, S Bosak, R Sequerra, A Fraser, SE Mango, WM Saxton, S Strome, S van den Heuvel, F Piano, J Vandenhaute, C Sardet, M Gerstein, L Doucette-Stamm, KC Gunsalus, JW Harper, ME Cusick, FP Roth, DE Hill, and M Vidal. A map of the interactome network of the metazoan c. elegans. *Science*, 303: 540–543, 2004.
76. Zhi Liang, Meng Xu, Maikun Teng, and Liwen Niu. NetAlign: a web-based tool for comparison of protein interaction networks. *Bioinformatics*, 22(17):2175–2177, 2006.
77. Chung-Shou Liao, Kanghao Lu, Michael Baym, Rohit Singh, and Bonnie Berger. Isorankn: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25(12):i253–258, 2009.
78. L.J. Lu, Y. Xia, A. Paccanaro, H. Yu, and M. Gerstein. Assessing the limits of genomic data integration for predicting protein networks. *Genome Res.*, 15:945–953, 2005.
79. V. Memisević, T. Milenković, and N. Pržulj. Complementarity of network and sequence information in homologous proteins. *Journal of Integrative Bioinformatics*, 7(3):135, 2010.
80. V. Memisević, T. Milenković, and N. Pržulj. An integrative approach to modeling biological networks. *Journal of Integrative Bioinformatics*, 7(3):120, 2010.
81. T. Milenković, J. Lai, and N. Pržulj. Graphcrunch: a tool for large network analyses. *BMC Bioinformatics*, 9(70), 2008.
82. T. Milenković, V. Memisević, A. K. Ganesan, and N. Pržulj. Systems-level cancer gene identification from protein interaction network topology applied to melanogenesis-related interaction networks. *Journal of the Royal Society Interface*, doi:10.1098/rsif.2009.0192, 2009.

83. T. Milenković and N. Pržulj. Uncovering biological network function via graphlet degree signatures. *Cancer Informatics*, 6:257–273, 2008.
84. Tijana Milenkovic, Weng Leong Ng, Wayne Hayes, and Nataša Pržulj. Optimal network alignment with graphlet degree vectors. *Cancer Informatics*, 9:121–137, 2010.
85. R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon. Superfamilies of evolved and designed networks. *Science*, 303:1538–1542, 2004.
86. R. Milo, S. S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298:824–827, 2002.
87. M. Molloy and B. Reed. A critical point of random graphs with a given degree sequence. *Random Structures and Algorithms*, 6:161–180, 1995.
88. M. Molloy and B. Reed. The size of the largest component of a random graph on a fixed degree sequence. *Combinatorics, Probability and Computing*, 7:295–306, 1998.
89. E Nabieva, K Jim, A Agarwal, B Chazelle, and M Singh. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, 21: i302–i310, 2005.
90. M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2): 167–256, 2003.
91. M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64:026118–1, 2001.
92. M. E. J. Newman and D. J. Watts. Renormalization group analysis in the small-world network model. *Physics Letters A*, 263:341–346, 1999.
93. M. E. J. Newman and D. J. Watts. Scaling and percolation in the small-world network model. *Physical Review E*, 60:7332–7342, 1999.
94. S.R. Paladugu, S. Zhao, A. Ray, and A. Raval. Mining protein networks for synthetic genetic interactions. *BMC Bioinformatics*, 9(426), 2008.
95. Jodi R Parrish, Jingkai Yu, Guozhen Liu, Julie A Hines, Jason E Chan, Bernie A Mangiola, Huamei Zhang, Svetlana Pacifico, Farshad Fotouhi, Victor J DiRita, Trey Ideker, Phillip Andrews, and Russell L Finley Jr. A proteome-wide protein interaction map for campylobacter jejuni. *Genome Biology*, 8:R130, 2007.
96. R. Pastor-Satorras, E. Smith, and R. V. Sole. Evolving protein interaction networks through gene duplication. *Journal of Theoretical Biology*, 222:199–210, 2003.
97. M. Penrose. *Geometric Random Graphs*. Oxford Univeristy Press, 2003.
98. Dmitry K. Pokholok, Christopher T. Harbison, Stuart Levine, Megan Cole, Nancy M. Hannett, Tong Ihn Lee, George W. Bell, Kimberly Walker, P. Alex Rolfe, Elizabeth Herbolsheimer, Julia Zeitlinger, Fran Lewitter, David K. Gifford, and Richard A. Young. Geome-wide map of nucleosome acetylation and metylation in yeast. *Cell*, 122:517–527, 2005.
99. N. Pržulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23:e177–e183, 2007.
100. N. Pržulj, D. G. Corneil, and I. Jurisica. Modeling interactome: Scale-free or geometric? *Bioinformatics*, 20(18):3508–3515, 2004.
101. N. Pržulj, D. G. Corneil, and I. Jurisica. Efficient estimation of graphlet frequency distributions in protein-protein interaction networks. *Bioinformatics*, 22(8):974–980, 2006. doi:10.1093/bioinformatics/btl030.
102. N. Pržulj and D. J. Higham. Modelling protein-protein interaction networks via a stickiness index. *Journal of the Royal Society Interface*, 3(10):711–716, 2006.
103. N. Pržulj, O. Kuchaiev, A. Stevanovic, and W. Hayes. Geometric evolutionary dynamics of protein interaction networks. *2010 Pacific Symposium on Biocomputing (PSB)*, 2010.
104. N. Pržulj, D. Wigle, and I. Jurisica. Functional topology in a network of protein interactions. *Bioinformatics*, 20(3):340–348, 2004.
105. P. Radivojac, K. Peng, W. T. Clark, B. J. Peters, A. Mohan, S. M. Boyle, and Mooney S. D. An integrated approach to inferring gene-disease associations in humans. *Proteins*, 72(3):1030–1037, 2008.
106. J.-D. Rain, L. Selig, H. De Reuse, V. Battaglia, C. Reverdy, S. Simon, G. Lenzen, F. Petel, J. Wojcik, V. Schachter, Y. Chemama, A. Labigne, and P. Legrain. The protein-protein interaction map of helicobacter pylori. *Nature*, 409:211–215, 2001.

107. O. Ratmann, C. Wiuf, and J. W. Pinney. From evidence to inference: probing the evolution of protein interaction networks. *HFSP Journal*, 2009. Published online 19 October 2009.
108. J.-F. Rual, K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, N. Li, G. F. Berriz, F. D. Gibbons, M. Dreze, N. Ayivi-Guedehoussou, N. Klitgord, C. Simon, M. Boxem, S. Milstein, J. Rosenberg, D. S. Goldberg, L. V. Zhang, S. L. Wong, G. Franklin, S. Li, J. S. Albala, J. Lim, C. Fraughton, E. Llamosas, S. Cevik, C. Bex, P. Lamesch, R. S. Sikorski, J. Vandenhaute, H. Y. Zoghbi, A. Smolyar, S. Bosak, R. Sequerra, L. Doucette-Stamm, M. E. Cusick, D. E. Hill, F. P. Roth, and M. Vidal. Towards a proteome-scale map of the human protein-protein interaction network. *Nature*, 437:1173–78, 2005.
109. M.P. Samanta and S. Liang. Predicting protein functions from redundancies in large-scale protein interaction networks. *PNAS*, 100:12579–12583, 2003.
110. A.S. Schwartz, J. Yu, K.R. Gardenour, Finley R.L. Jr., and T. Ideker. Cost-effective strategies for completing the interactome. *Nature Methods*, 6(1):55–61, 2009.
111. B. Schwikowski, P. Uetz, and A. Fields. A network of protein-protein interactions in yeast. *Nature Biotechnology*, 18:1257–1261, 2000.
112. R. Sharan, T. Ideker, B. P. Kelley, R. Shamir, and R. M. Karp. Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. In *Proceedings of the eighth annual international conference on Computational molecular biology (RECOMB'04)*, 2004.
113. R. Sharan, I. Ulitsky, and R. Shamir. Network-based prediction of protein function. *Molecular Systems Biology*, 3(88):1–13, 2007.
114. Roded Sharan and Trey Ideker. Modeling cellular machinery through biological network comparison. *Nature Biotechnology*, 24(4):427–433, Apr 2006.
115. Roded Sharan and Trey Ideker. Protein networks in disease. *Genome Research*, 18:644–652, 2008.
116. Roded Sharan, Silpa Suthram, Ryan M. Kelley, Tanja Kuhn, Scott McCuine, Peter Uetz, Taylor Sittler, Richard M. Karp, and Trey Ideker. Conserved patterns of protein interaction in multiple species. *Proceedings of the National Academy of Sciences of the United States of America*, 102(6):1974–1979, 2005.
117. S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics*, 31:64–68, 2002.
118. H. A. Simon. On a class of skew distribution functions. *Biometrika*, 42:425–440, 1955.
119. Nicolas Simonis, Jean-Francois Rual, Anne-Ruxandra Carvunis, Murat Tasan, Irma Lemmens, Tomoko Hirozane-Kishikawa, Tong Hao, Julie M. Sahalie, Kavitha Venkatesan, Fana Gebreab, Sebiha Cevik, Niels Klitgord, Changyu Fan, Pascal Braun, Ning Li, Nono Ayivi-Guedehoussou, Elizabeth Dann, Nicolas Bertin, David Szeto, Ameli Dricot, Muhammed A. Yildirim, Chenwei Lin, Anne-Sophie De Smet, Huey-Ling Kao, Christophe Simon, Alex Smolyar, Jin Sook Ahn, Muneesh Tewari, Mike Boxem amd Stuart Milstein, Haiyuan Yu, Matija Dreze, Jean Vandenhaute, Kristin C. Gunsalus, Michael E. Cusick, David E. Hill, Jan Tavernier, Frederick P. Roth, and Marc Vidal. Empirically controlled mapping of the caenorhabditis elegans protein-protein interactome network. *Nature Methods*, 6(1):47–54, 2009.
120. R. Singh, J. Xu, and B. Berger. Pairwise global alignment of protein interaction networks by matching neighborhood topology. In *Research in Computational Molecular Biology*, pages 16–31. Springer, 2007.
121. R. Singh, J. Xu, and B. Berger. Global alignment of multiple protein interaction networks. *Proceedings of Pacific Symposium on Biocomputing 13*, pages 303–314, 2008.
122. C. Song, S. Havlin, and H. A. Makse. Self-similarity of complex networks. *Nature*, 433: 392–395, 2005.
123. U. Stelzl, U. Worm, M. Lalowski, C. Haenig, F.H. Brembeck, H. Goehler, M. Stroedicke, M. Zenkner, A. Schoenherr, S. Koeppen, J. Timm, S. Mintzlaff, C. Abraham, N. Bock, S. Kietzmann, A. Goedde, E. Toksoz, A. Droege, S. Krobitsch, B. Korn, W. Birchmeier, H. Lehrach, and E.E. Wanker. A human protein-protein interaction network: A resource for annotating the proteome. *Cell*, 122:957–968, 2005.

124. M. P. H. Stumpf, C. Wiuf, and R. M. May. Subnets of scale-free networks are not scale-free: Sampling properties of networks. *Proceedings of the National Academy of Sciences*, 102:4221–4224, 2005.
125. S. Suthram, T. Sittler, and T. Ideker. The plasmodium protein network diverges from those of other eukaryotes. *Nature*, 438:108112, 2005.
126. R. Tanaka. Scale-rich metabolic networks. *Physical Review Letters*, 94:168101, 2005.
127. A. H. Y. Tong, G. Lesage, G. D. Bader, H. Ding, H. Xu, X. Xin, J. Young, G. F. Berriz, R. L. Brost, M. Chang, Y. Chen, X. Cheng, G. Chua, H. Friesen, D. S. Goldberg, J. Haynes, C. Humphries, G. He, S. Hussein, L. Ke, N. Krogan, Z. Li, J. N. Levinson, H. Lu, P. Menard, C. Munyana, A. B. Parsons, O. Ryan, R. Tonikian, T. Roberts, A.-M. Sdicu, J. Shapiro, B. Sheikh, B. Suter, S. L. Wong, L. V. Zhang, H. Zhu, C. G. Burd, S. Munro, C. Sander, J. Rine, J. Greenblatt, M. Peter, A. Bretscher, G. Bell, F. P. Roth, G. W. Brown, B. Andrews, H. Bussey, and Charles Boone. Global mapping of the yeast genetic interaction network. *Science*, 303:808–813, 2004.
128. P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, E. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleish, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, and J. M. Rothberg. A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae. *Nature*, 403:623–627, 2000.
129. Peter Uetz, Yu-An Dong, Christine Zeretzke, Christine Atzler, Armin Baiker, Bonnie Berger, Seesandra Rajagopala, Maria Roupelieva, Dietlind Rose, Even Fossum, and Jrgen Haas. Herpesviral protein networks and their interaction with the human proteome. *Science*, 311:239–242, 2006.
130. O. Vanunu, O. Magger, E. Ruppin, T. Shlomi, and R. Sharan. Associating genes and protein complexes with disease via network propagation. *PLoS Computational Biology*, 6:e1000641, 2010.
131. A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani. Modeling of protein interaction networks. *ComPlexUs*, 1:38–44, 2001.
132. A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani. Global protein function prediction from protein-protein interacton networks. *Nature Biotechnology*, 21:697–700, 2003.
133. K. et al. Venkatesan. An empirical framework for binary interactome mapping. *Nature Methods*, 6(1):83–90, 2009.
134. Albrecht von Brunn1, Carola Teepe, Jeremy C. Simpson, Rainer Pepperkok, Caroline C. Friedel, Ralf Zimmer, Rhonda Roberts, Ralph Baric, and Jurgen Haas. Analysis of intraviral protein-protein interactions of the sars coronavirus orfeome. *PLoS ONE*, 2:e459, 2007.
135. C. von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields, and P. Bork. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417(6887):399–403, 2002.
136. S. Wachi, K. Yoneda, and R. Wu. Interactome-transcriptome analysis reveals the high centrality of genes differentially expressed in lung cancer tissues. *Bioinformatics*, 21: 4205–4208, 2005.
137. A. Wagner. How the global structure of protein interaction networks evolves. *Proceedings of The Royal Society of London. Series B, Biological Sciences*, 270:457–466, 2003.
138. D. J. Watts and S. H. Strogatz. Collective dynamics of ‘small-world’ networks. *Nature*, 393:440–442, 1998.
139. D. B. West. *Introduction to Graph Theory*. Prentice Hall, Upper Saddle River, NJ., 2nd edition, 2001.
140. D. S. Wishart, C. Knox, A. C. Guo, S. Shrivastava, M. Hassanali, P. Stothard, Z. Chang, and J. Woolsey. Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Research*, 34:D668–D672, 2006.
141. SJ Wodak, S Pu, J Vlasblom, and B Seraphin. Challenges and rewards of interaction proteomics. *Mol. Cell Proteomics*, 8(1):3–18, 2009.

142. I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S. M. Kim, and Eisenberg D. DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Research*, 30(1):303–305, 2002.
143. E Yeger-Lotem, S Sattath, N Kashtan, S Itzkovitz, R Milo, RY Pinter, U Alon, and H Margalit. Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction. *Proc Natl Acad Sci U S A*, 101(16):5934–5939, April 2004.
144. H. et al. Yu. High-quality binary protein interaction map of the yeast interactome networks. *Science*, 322:104–110, 2008.
145. M. Zaslavskiy, F. Bach, and J. P. Vert. Global alignment of protein-protein interaction networks by graph matching methods. *Bioinformatics*, 25(12):i259–i267, 2009.

Chapter 3
Function Annotation in Gene Networks

Petko Bogdanov, Kathy Macropol, and Ambuj K. Singh

Abstract Modern sequencing technology enables the discovery of new gene products in an increasing number of organisms. However, the sequence on its own does not provide sufficient information about cellular mechanisms and their function. Efforts need to be directed toward genome characterization at the molecular level. Wet-lab experiments in this direction are assisted by a variety of computational methods that exploit the abundance of data.

The advent of high-throughput interaction detection methods has generated large amounts of gene interaction data. This has allowed the construction of genome-wide networks. Studying genomes in a networked setting has been beneficial for global annotation in two ways. First, there has been an increasing number of network-based function prediction methods. Second, networks have inspired the community to revisit the definition of gene function. The original molecular characterization of function has been extended to a multi-molecule function, termed *biological process* [Gene ontology: Tool for the unification of biology. Nature, 2000] in recently emerging annotation systems.

In this chapter, we present the current methods of automated annotation of protein functions. We describe existing annotation prediction methods and ontologies used to define a gene's function at the molecular and process level. We discuss in detail the workings of a generalized framework for network prediction and present experimental accuracy comparison of several popular methods within this framework. We also discuss the use of networks from multiple species for annotation enrichment in sparse genomes.

P. Bogdanov (✉)
Department of Computer Science, University of California,
Santa Barbara, CA 93106, USA
e-mail: petko@cs.ucsb.edu

M. Koyutürk et al. (eds.), *Functional Coherence of Molecular Networks in Bioinformatics*, DOI 10.1007/978-1-4614-0320-3_3,
© Springer Science+Business Media, LLC 2012

1 Computational Function Prediction and Gene Networks

The rapid development of genomics and proteomics has generated an unprecedented amount of data for multiple model organisms. As has been commonly realized, the acquisition of data is but a preliminary step, and a true challenge lies in developing effective means to analyze such data and endow them with physical or functional meaning [54]. Annotation systems attempt to systematically characterize all relevant gene aspects in a human-friendly and consistent manner that also enables automated processing.

Traditionally, gene function has been associated with the molecular properties of the sequence, such as catalytic activity, signalling activity, and others. The established approach to infer such properties for newly discovered genes combines sequence/structure homology and manual verification in the wet lab. The first step, referred to as computational function prediction, facilitates the functional annotation by directing the experimental design to a narrow set of possible annotations for unstudied genes. The information that initial computational methods considered was limited to the sequence and structure, independent of other molecules and gene products that act jointly with the target sequence.

Recently, a different type of high-throughput data has been employed for the automated annotation effort. Methods like microarray coexpression analysis and yeast two-hybrid experiments have allowed the construction of large interaction networks. An interaction network consists of nodes representing genes and edges representing interactions between genes. Due to the inherent uncertainty of input data sources, networks are often stochastic, with edges weighted by the probability of interaction. There is more information in a network compared to sequence or structure alone. A network provides a global view of the context of each gene. Hence, the next challenge of computational function prediction is the use of a gene's interaction context within the network to predict its function.

With the realization that interactions encode functional dependencies, the notion of function has been expanded to activities performed by multiple gene products in conjunction. Annotation formats have been accordingly revised and new annotation types have been added. The addition of genomic context and the abundant interaction data repositories, as compared to sequence and structure data alone, has led to a new paradigm of multi-level specificity and multi-modal gene annotations.

A node in an interaction network is annotated with one or more annotation terms. Multiple and sometimes unrelated annotations can occur due to multiple active binding sites or possibly multiple stable tertiary conformations of a corresponding gene product [26]. The annotation terms are commonly based on an ontology. A major effort in this direction is the Gene Ontology (GO) project [4]. GO characterizes genes in three major aspects: *molecular function*, *biological process*, and *cellular localization*. *Biological process* captures the higher level multi-molecule functions, while *molecular function* describes the individual biochemical properties of a sequence or a complex. Since the project initiation, GO has been widely used for the annotation of studied genomes, as it addresses the limitations of previous

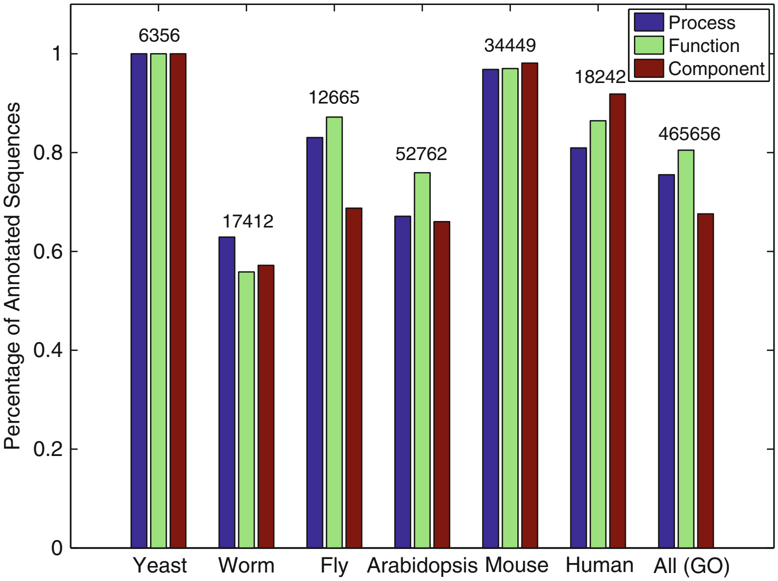


Fig. 3.1 GO annotations in several model organisms. Numbers denote the total number of known sequences for a given organism

annotation systems due to its explicit differentiation of processes from function and the addition of cellular component as an intrinsic gene product feature.

The state of GO annotation availability as of May 2010 is presented in Fig. 3.1. Approximately, 30% of worm gene products are uncharacterized, although it has been one of the most popular model systems over the years. It is important to note that even though the percentage of unknown gene products may seem small for some genomes, this statistic is somewhat optimistic. First, there are gene products that perform multiple molecular functions, participate in multiple processes or can be found in different cell compartments. Hence, the existence of at least one annotation for each yeast gene product does not mean that this genome is fully annotated. Second, a significant number of genes are characterized by general terms belonging to the first levels of the GO hierarchy. The specificity of a gene's annotations translates directly to the amount of knowledge about the actual function of the gene. In this respect, many of the available annotations need to be improved by making them more specific. The methods we discuss in the remainder of this chapter address both of the above challenges.

The annotation of a gene's function is an ongoing process and computational approaches are actively used to assist experimentation. The rate at which interaction data is generated is growing steadily and this phenomenon has established network-based annotation methods as a natural extension and complement to homology-based annotation. To quantify this rapid growth, more than 1,000 microarray experiments (a prominent interaction inference source) were made publicly available through the *Gene Expression Omnibus* (GEO) repository in 2008, and this number has been steadily growing since 2002. Note that this is only one of the online repositories for microarray data, and microarray analysis is only one of the multiple high-throughput methods traditionally used to infer interactions among genes. Aggregating all available interaction data can only improve computational annotation methods, relying on interaction networks.

In the following sections of this chapter, we discuss existing methods of network composition and the specifics of annotation schemes with a focus on GO (Sect. 2). Understanding network and annotation origins is a necessary preliminary step to exploring the space of available network-based annotation approaches, since the applicability and effectiveness of any approach is dependent both on the target annotation type and the nature of the underlying interaction network. We map the space of existing methods in Sect. 3. We generalize the problem of network-based annotation in Sect. 4 and present performance comparison for several recent methods in Sect. 5. Finally, we discuss ideas on tandem prediction using multiple organism networks (Sect. 6) and homology information.

2 Network Synthesis and Annotation

Currently, a significant amount of data on gene product interactions are obtained through the use of high throughput methods. Many of these diverse data sources have been integrated to produce functional association networks [29, 30].

Protein–protein physical interactions, which reflect the functional associations of their corresponding genes, can be inferred by a variety of methods. *Yeast-two-hybrid* assays discover protein pairs that physically interact, based on transcriptional activation. Another technique in this category is based on mass spectroscopy. Analysis of molecule spectra provides an efficient tool for identifying proteins composing a protein complex after purification. In both cases, the corresponding interaction data contain false positives, and methods to assess the reliability of interactions have been proposed [14].

Genetic interactions represent a different kind of gene association, traditionally detected through genome-wide synthetic lethal screens [20]. A synthetic lethal reaction between two genes occurs when mutations to the genes are nonlethal separately, but result in cell death when occurring together. A synthetic lethal reaction reveals a genetic interaction between the corresponding gene products. Experimental detection of genetic interactions can be cumbersome and expensive, depending on the model organism. Computational alternatives have been developed for their inference [11, 51].

Another kind of gene association, termed functional interactions, may be discovered from diverse sources such as microarray coexpression analysis, phylogenetic profiling, literature mining, and others. Microarray datasets are publicly available from repositories such as the GEO [2]. The correlation of expression levels for pairs of genes from multiple experimental conditions is used as indicator signal of their functional linkage. Aggregating correlations from large number of coexpression assays allows the building of a genome-wide functional interaction networks [50]. The resulting networks can be edge-weighted based on the strength of correlation between the expression levels of the corresponding genes.

Besides microarrays, another source of functional interaction data is available due to *phylogenetic profiling* [33]. This interaction inference approach detects joint presence or absence of a pair of genes across multiple organisms to infer a functional connection. Literature mining has also been employed to predict linkage between genes [23]. Co-occurrence of gene names is used to estimate the likelihood that the respective genes have a functional linkage.

There have been multiple functional annotation schemes introduced over the years, with several used at present. Annotation terms corresponding to sequence function are associated with gene products. Systematic application of these schemes has resulted in the accumulation of organism-specific annotation databases that can easily be analyzed and understood. In most systems, the annotations are arranged as hierarchies, in which terms located higher refer to general functions, and ones occupying lower levels refer to increasingly specific functions. The first such extensive method, the Riley scheme [41], was introduced in 1993 to categorize *Escherichia coli* genes. Multifun [42], a later modification to the approach, allowed for the mapping of multiple functional categories per gene. FunCat [44] expanded upon previous schemes by allowing hierarchical annotations to be applied to a number of organisms [18, 38, 43, 45].

A recent system, which varies from the traditional tree structure, is the GO [4]. According to GO, annotation terms may have multiple parent terms, forming a Directed Acyclic Graph (DAG). In addition, there are various relations associated with every directed edge, such as “is a subset of” or “is a part of.” These additions allow for greater flexibility in categorizing gene products.

GO defines three annotation domains: molecular function, biological process, and cellular compartment. Molecular functions describe activities performed by individual gene products at the molecular level. Biological processes describes the operations with a defined start and end, performed by groups of interacting gene products and pertinent to the functioning of cells, tissues, organs, and whole organisms. GO is the first gene annotation ontology that distinguishes between the two widely used and distinct notions of gene function: individual molecular function, determined mostly by the sequence and structure, and group (global) function, characterized by the interaction of multiple gene products and termed process in GO. The third domain, cellular component, characterizes a specific cellular or extracellular environment in which a gene product operates.

The GO annotation scheme has become widely used since its introduction. It has been applied to 49 species at present and this number can only continue to grow due to the generality of the categories and its inherent extensibility. There are over 30,000 GO annotation terms, with the majority of them (62%) belonging to the biological process category [3].

3 Techniques for Computational Annotation in Networks

Interaction networks provide a systematic view of whole genomes, and hence the synthesis of the first genome-wide maps has been followed by attempts to discover and predict function within the system as a whole. It is important to realize that network-based prediction methods are highly dependent on the semantics and uncertainty of the links. Most high-throughput interaction experiments contain noise due to the nature of their method. Prediction methods should ideally take this into account. In this respect, the interpretation of interaction data for the purposes of network synthesis should be an obligatory first step when defining an annotation prediction scheme.

Some interaction data sources are highly correlated with each other while others are near orthogonal. The semantic of a physical interaction link, for example, is quite different from that of a *synthetic sick or lethal* interaction. Recent genome-wide studies have demonstrated that genetic interactions tend to run orthogonal to physical interactions [11, 51]. Physical interactors complement each other in performing a common biological process, while genetic interactors indicate genes that partially overlap in functionality and can back up one another.

In this section, we describe the general concepts of annotation prediction algorithms, grouped by their main hypothesis and the computational tools that comprise them. We provide additional discussion on the interplay between network synthesis and the prediction algorithms.

A study from 2007 [48] groups most original network-based prediction methods into several groups: *module-assisted*, *direct methods*, and *probabilistic methods*. Direct methods seek to transfer annotations along the network edges, module-assisted ones detect network modules and transfer functions within them, while probabilistic methods build a graphical model of the network that capture annotation co-occurrences, later used for global prediction.

3.1 Direct Approaches

A common hypothesis among direct methods is that interacting genes have similar annotations. This is why they are very successful in predicting biological processes in physical and functional interaction networks. One of the earliest approaches called *Majority* [47] predicts the prevailing annotations among the direct interactors of a target gene. This idea has later been generalized to higher levels in the network [22]. Both of these methods are effective when the interaction neighborhood of the target gene is not sparse in annotations. The *Functional Flow* method [39] simulates a network flow of annotations from annotated genes to target ones. Karaoz et al. [24] propose an annotation technique that maximizes the number of edges between genes with the same function.

3.2 Module-Assisted Approaches

A second major group, called module-assisted methods, exploits the modularity in biological networks, where modules correspond to cellular complexes. The members of this group detect network modules and then perform module-wide annotation enrichment [34], following the assumption that all genes within a network module have the same annotation. Similar to direct methods, module-assisted approaches are effective in predicting *biological processes* in physical interaction maps.

Methods comprising the module-assisted group differ in the manner in which they identify modules. Some use graph clustering [17, 35, 49, 55], while others use hierarchical clustering based on network distance [6, 8, 34] and common interactors [46]. The clustering approach by Macropol and colleagues [35] was shown to outperform well-established clustering schemes in identifying known protein clusters. Although the method was not proposed as a function prediction technique, its superior cluster accuracy makes it an advantageous ingredient of a module-assisted annotation technique.

A common premise of both direct and module-assisted methods is that genes with similar annotations are always topologically close in the network. As Fig. 3.2 shows, the molecular function annotations in gene networks do not corroborate this hypothesis. The direct methods are also limited to utilize information about neighbors up to a certain level. Thus, their accuracy is dependent on the density of annotations in the interaction context.

3.3 Probabilistic Approaches

A different group of techniques considers probabilistic models based on *Markov Random Fields* [15, 16, 32, 53]. This set of predictors seeks to compute annotation for all network nodes at once, while optimizing a global optimization criterion. The main driving principle of these techniques is that a target gene annotation is independent of all other genes, given its neighbors [48]. Estimates of prior and conditional probabilities of annotations are first computed followed by the joint likelihood of all target annotations. Probabilistic network analysis has also been used for disease gene identification [28, 52].

All three groups discussed above associate genes that are direct or close neighbors with similar annotations. While this is effective for biological process prediction, genes of the same molecular function can be far from each other in the network (Fig. 3.2). Furthermore, particular molecular functions are surrounded by patterns of different molecular functions within a cellular module, as they complement each other in performing a certain process.

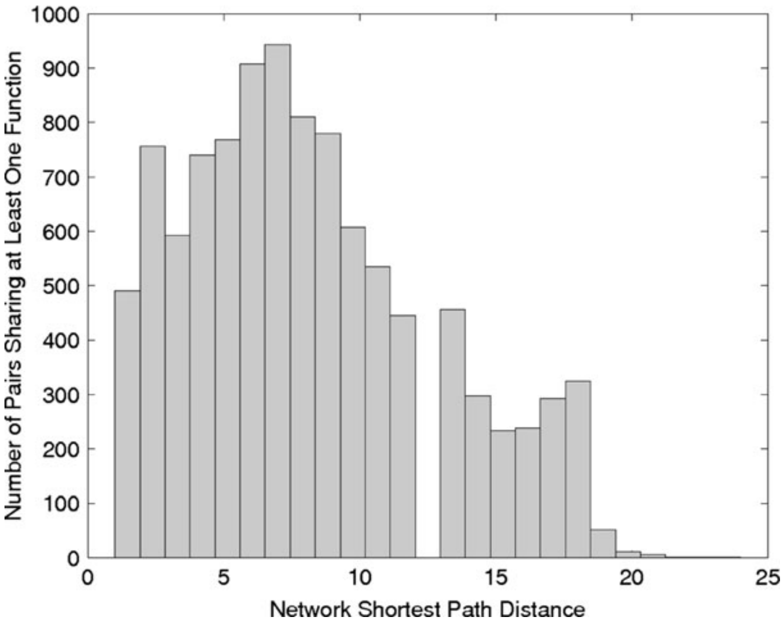


Fig. 3.2 Genes sharing annotations do not always interact in the *filtered yeast interactome* (FYI) [19]. Similar functions are sometimes at large network distances. The summary is based on all pairs of annotated genes

3.4 Pattern-Based Approaches

A recent body of work revolves around the idea of network patterns for function prediction. One approach that can be classified in this category is *Indirect Neighbor* [12]. The authors' hypothesis is that genes that share interaction neighbors perform a similar function as well. The method distinguishes between direct and indirect (second level) functional associations and applies an appropriate weighting between the two. In a sense, this method is a mixture between a direct method and a pattern-based approach.

A purely network pattern technique called *LaMoFinder* [10] predicts annotations based on network structure motifs. An unannotated network is first mined for conserved and unique structural patterns called motifs. Pairs of corresponding genes in different motif occurrences are expected to have similar annotations. The method is restricted to target genes that are part of unique and frequent structural motifs. A less conservative approach for pattern extraction is proposed in [27]. According to this method, a pattern is a rooted subgraph around a target gene. The authors define a similarity measure between annotations of patterns and predict a target annotation based on its most similar patterns. Both pattern-based approaches rely on exact or relaxed topological matches of subgraphs. High-throughput interaction detection,

Table 3.1 Type of network prediction methods and their corresponding underlying hypothesis

Methods	Algorithm type	Hypothesis
Majority [47], Hishigaki et al. [22], Karaoz et al. [24], Functional flow [39]	Direct	Annotation clustering
Macropol et al. [35], Dunn et al. [17], Zhang et al. [55], Spirin et al. [49], Arnau et al. [6], Brun et al. [8], Samanta et al. [46]	Module-assisted	Annotation clustering
Deng et al. [15, 16], Letovsky et al. [32], SBIA [53]	Probabilistic	Annotation clustering
Indirect neighbor [12], *LaMoFinder* [10], Kirac et al. [27], Bogdanov et al. [7]	Pattern-based	Similar neighborhood

however, produces inherently uncertain data. Hence, probabilistic techniques are needed to analyze the network topology and define network patterns.

A recent method [7] hypothesizes that the simultaneous activity of sometimes functionally diverse functional agents comprise higher level processes in different regions of the network. The authors refer to this hypothesis as *Similar Neighborhood* and to the principle in all direct, module-assisted and probabilistic methods as *Annotation Clustering*. Justification for *Similar Neighborhood* is provided in Fig. 3.2, which reveals that genes of similar function may occur at large network distances. Driven by this principle, the authors seek to predict unknown annotations based on similar functional neighborhood patterns. Functional network neighborhoods are first extracted from the interaction network and then used to train an annotation classifier. We delve into the method in more detail in the next section.

To summarize, we categorize the existing network-based annotation predictors in four groups: *direct, module-assisted, probabilistic, and pattern-based*. The first three are motivated by the *Annotation Clustering* hypothesis, postulating that similar annotations are more likely to interact than diverse ones. Different from this premise, pattern-based approaches follow the *Similar Neighborhood* principle. The accuracy of each of these methods in a prediction task depends on the type of input network and the annotation type that is sought for prediction. A summary of the discussed methods is presented in Table 3.1.

4 Network Annotation Prediction as a Framework

We generalize the process of network-based annotation prediction as a three step framework: (1) interaction network synthesis (2) network feature extraction, and (3) classification. The first step concerns the processing and interpretation of interaction data for the purposes of building a genome-wide interaction map. The second step summarizes relevant information for a target node which is then used to predict annotations in the third step. Many of the approaches from Sect. 3 fit implicitly in this three-step model, although the corresponding papers focus on some steps more than others. The latter two steps were proposed by Bogdanov and colleagues [7].

In this section, we will follow the development of the approach in [7] and will provide additional discussion on how other methods can be structured according to the three-step framework.

The proposed generalization has several advantages. First, it explicitly decouples the tasks comprising a prediction method and thus allows for the possibility that other kinds of network synthesis can be performed, other neighborhood features can be extracted, and that other kinds of classifiers can be used. Second, the separation in steps may enable combination of different known techniques and algorithms in any of the phases, resulting in composite approaches. In addition, performance analysis will be more comprehensive if each of the steps are assessed separately. Note that the third step is a central problem in data mining, and hence any of the state of the art classification schemes can be employed.

The method proposed by Bogdanov and colleagues [7] is based on the *Similar Neighborhood* hypothesis. It uses functional networks, synthesized according to the same principles as previous competing methods [39]. Given a network, the authors summarize the functional context of a target gene in the neighborhood feature extraction step. The method computes the steady state distribution of a *Random Walk with Restarts (RWR)* from the gene. The steady state is then transformed into a functional profile. In the third step, *Neighborhood Patterns* employs a *K-Nearest-Neighbors (KNN)* classifier to predict the function of a target gene based on its functional profile. As confirmed by the experimental results accompanying the method, the desired trade-off between accuracy of prediction and coverage of can be controlled by *k*, the only parameter of the KNN classification scheme.

4.1 Interaction Network Synthesis

Employing a single interaction source for annotation prediction is prone to false positives and incomplete genome coverage due to high-throughput methods. To overcome these limitations, aggregation and fusion of multiple interaction evidence sources into a single network has been employed in a number of studies [29–31]. By integrating the various methods discussed in Sect. 2, a more accurate and complete picture of the interactome can be obtained.

To synthesize a combined functional network, all data sources may be probabilistically combined. Every interaction discovered through a source adds strength to the evidence of existence for the specific linkage. The resulting network is characterized by a weight on every edge corresponding to the level of the confidence in the interaction. There are several methods that propose to combine multiple sources of interaction evidence. One such approach, developed in [29–31], assigns a *log-likelihood* score to the interaction from any single data source. The score reflects the probability of the linkage, and is calculated using a Bayesian method incorporating prior knowledge from a gold standard. Log-likelihood scores are then combined using a weighted sum to form the final genome-wide network.

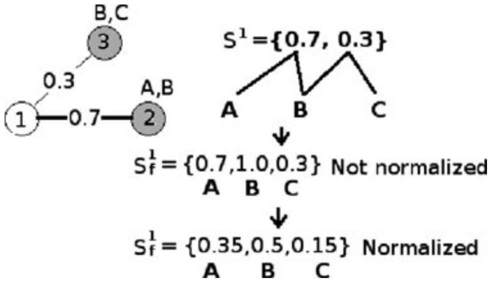

B,C
3
0.3
A,B
1
0.7
2
S1={0.7, 0.3}
A B C
Sf1= {0.7,1.0,0.3} Not normalized
A B C
Sf1= {0.35,0.5,0.15} Normalized
A B C

Fig. 3.3 Transformation of the neighborhood profile of node 1 into a functional profile. Node 2 is annotated with functions A and B and node 3 is annotated with functions B and C. The neighborhood profile of node 1 is computed and transformed using the annotations on the nodes into a functional profile

4.2 Network Feature Extraction

The authors of the method proposed in [7] extract network features in two steps. First, the neighborhood of a target node is characterized with respect to all other nodes in the network. Second, this node-based characterization is transformed to a function-based one.

A gene's neighborhood is first summarized by computing the steady state distribution of a *RWR*. The trajectory of a random walker that starts from the target gene and moves to its neighbors with a probability proportional to the weight of each connecting edge is simulated. The random walker is kept close to the original node to explore its local neighborhood, by allowing transitions to the original node with a probability of r, the restart probability [9].

The network graph is represented by its adjacency matrix $M_{n,n}$. An element $m_{i,j}$ of M encodes the probability of an interaction between genes i and j. The outgoing edge weights of each gene are normalized. Let us term the steady state distribution of node j as the *neighborhood profile* of gene j, and denote it as $S^j, j \in [1,n]$. The neighborhood profile is a vector of probabilities S_i^j, $i \neq j$, $i, j \in [1,n]$. Component S_i^j is proportional to the frequency of visits to node i in the RWR from j. Solving for the stationary distribution can be done using a power iteration approach defined as follows:

$$S^j(t+1) = (1-r)M^T S^j(t) + rX. \tag{3.1}$$

In the above equation, X is a size-n vector defining the initial state of the random walk. In the above scenario, X has only one nonzero element corresponding to the target node. $S^j(t)$ is the neighborhood profile after t time steps. The final neighborhood profile is the vector S^j when the iteration converges. One interpretation of the neighborhood profile is that it defines an affinity vector of the target node to all other nodes based solely on the network structure.

The next step in feature extraction is the transformation of a neighborhood profile into a functional profile. The affinity S_i^j of node j to node i can be treated as affinity to the annotations of i. Figure 3.3 illustrates the transformation of a neighborhood profile to a functional profile. Assume that RWR performed from node 1 results in the neighborhood profile $(0.7, 0.3)$, where 0.7 corresponds to node 2, and 0.3 to node 3. Annotations on these two nodes are weighted by the corresponding values, resulting in the vector $(0.7, 1.0, 0.3)$ over functions A, B, and C, respectively. This vector is then normalized, resulting into the functional profile $(0.35, 0.5, 0.15)$.

More formally, based on the annotations of a gene, an annotation flag e_{ia} is set to 1 if gene i is annotated with function a and 0 otherwise. The affinity to each function a in the neighborhood profile is then computed as:

$$S_f^j(a) = \sum_{i=1, i \neq j}^{n} S_i^j e_{ia}. \tag{3.2}$$

Vector S_f^j is normalized to yield the functional profile for node j.

The outlined approach for feature extraction represents one way to summarize the interaction neighborhood of a gene. Interestingly, other methods, discussed in Sect. 4 can also be thought of extracting features and further use those for annotation prediction. *Direct* methods count annotations in direct or indirect neighbors; *module-assisted* represent each gene as part of a network cluster based on its connectivity; *probabilistic* methods compute conditional annotation probabilities based on neighboring genes; and *pattern-based* approaches associate genes with structural patterns in which they appear. All these signals can be interpreted as network-based features associated with each gene.

4.3 Classification

The function extraction step above produces a *functional profile* for each gene, summarizing a gene's affinity to annotations. According to the *Similar Neighborhood* hypothesis adopted in [7], genes with similar functional profiles are expected to have similar annotations. The method proceeds by classifying genes using a k Nearest Neighbor classifier and an *Manhattan (L_1)* distance metric to quantify the dissimilarity of two functional profiles.

The consensus set of predicted annotations is computed using weighted voting. Annotations of a more similar neighborhood are weighted higher. The result is a set of scores for each function where a function's score is computed as follows:

$$F_a^j = \sum_{i=1}^{k} f(d(i,j)) e_{ia}, \tag{3.3}$$

where e_{ia} is an indicator value set to 1 if gene i is annotated with a, $d(i,j)$ is the distance between functional profiles of genes i and j and $f(d(i,j))$ is a function that transforms the distance to score. A distance-decreasing function of the form $f(d) = \frac{1}{1+\alpha d}, \alpha = 1$ is used for distance to score transformation.

The classification step can be implemented using any classification algorithm from the machine learning domain. In addition, this step is not limited to the use of functional neighborhood features only. As we discussed in the previous section, most existing methods compute a specific network-based representation of a gene. The actual prediction in all methods is equivalent to performing classification using the corresponding representation. Treating all approaches as classification techniques opens the possibility of applying more elaborate algorithms in the prediction phase as opposed to simple rule-based predictors.

5 Accuracy Comparison of Existing Methods

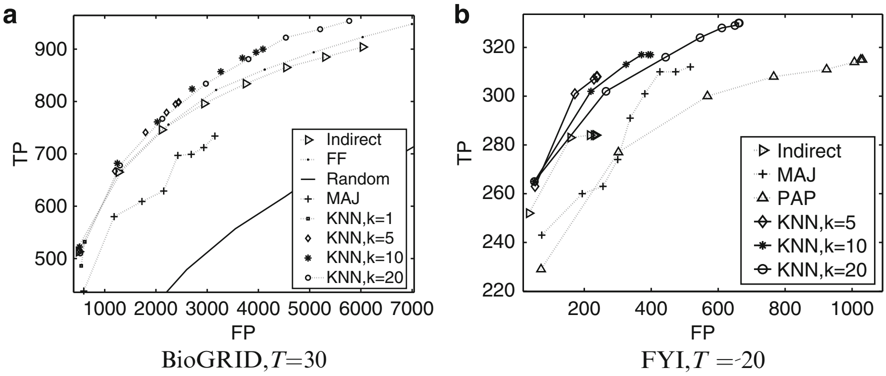
We next present performance evaluation, reported by Bogdanov and colleagues [7]. The authors compare a number of network-based methods on two yeast interaction networks. One of the networks used for evaluation is a high-confidence interaction network, termed *Filtered Yeast Interactome (FYI)* [19]. This network is synthesized by using a collection of interaction data sources, including high-throughput yeast two-hybrid, affinity purification and mass spectrometry, *in silico* computational predictions of interactions, and interaction complexes from MIPS [37]. The second yeast evaluation network is constructed by combining nine interaction data sources with genetic interactions from the *BioGRID* repository [5]. The method of construction is similar to the ones used in [12, 36, 39].

In this specific evaluation study, the first step of network synthesis is fixed for all competing techniques for the purpose of fair comparison. Note that for different data sets the performance of some techniques might change.

The methods used for comparison include the technique proposed by Bogdanov et al. [7], referred to as *KNN* in the experiments; two direct methods: *Majority (MAJ)* [47] and *Functional Flow (FF)* [39]; two pattern-based approaches, namely *Indirect Neighbors (Indirect)* [12] and *PAP* [27]; and a probabilistic method called *Statistically Based Iterative Algorithm (SBIA)* proposed in [53]. The various techniques are compared by performing leave-one-out validation experiments. Since the competing techniques implicitly use all available annotations, leave-one-out provides a fair comparison. In this setup, a target gene is held out (i.e., its annotations are considered unknown) and a prediction is computed using the rest of the annotation information in the network. All competing methods compute a score distribution for every class. The scores are used to rank the candidate annotations and the accuracy is analyzed for different ranks. An ideal technique would rank the true (held-out) annotation(s) highest.

A true positive (TP) prediction is a gene predicted as its actual label or any of the label's ontological descendants according to GO. This is also known as the *true path* prediction criterion and has been used in previous ontology-aware prediction studies [13]. Only frequent enough annotations are considered for comparison and the minimum threshold for the frequency is denoted as T.

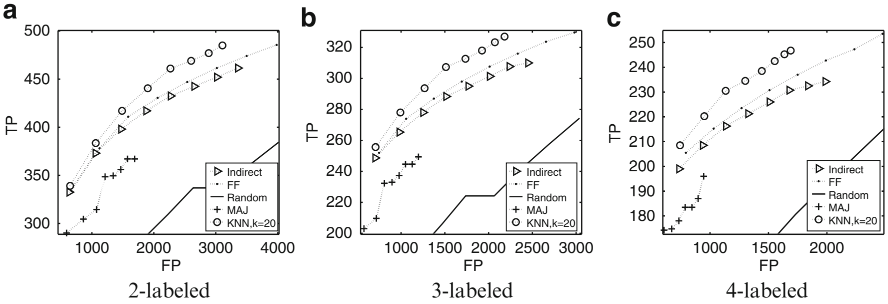
Figure 3.4 presents the TP versus false positive (FP) comparison when predicting genes of single known function. The KNN approach dominates the rest of the



BioGRID,T=30

FYI,T =20

Fig. 3.4 TP versus FP for single-labeled genes in the (**a**) *BioGRID* and (**b**) *FYI* networks



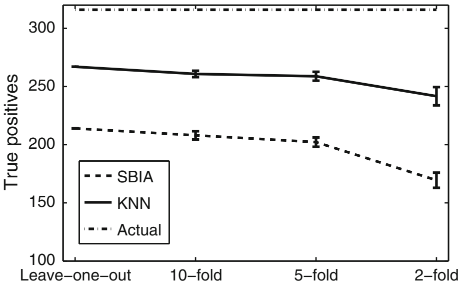
2-labeled

3-labeled

4-labeled

Fig. 3.5 Performance comparison on the *BioGRID* network for (**a**) 2-, (**b**) 3- and (**c**) 4-labeled genes, $T = 30$

Fig. 3.6 Accuracy of KNN compared to the probabilistic approach *SBIA* (*FYI*, $T = 20$, $k = 10$)



approaches, and all approaches are significantly better than a random predictor using the frequency of annotations for prediction. The performance comparison on multi-labeled genes is presented in Fig. 3.5. For this experiment, genes are grouped by the cardinality of their label set and leave-one-out validation is similarly performed.

A comparison with the probabilistic method *SBIA* [53] is presented in Fig. 3.6. *SBIA* is different from the above methods in that it performs collective classification.

Multiple targets are classified simultaneously and an assignment that maximizes the global likelihood is computed. It is compared with *kNN* using cross-validation, where the folds are sampled uniformly from the single-label annotated nodes in the network.

6 Annotation in Multiple Genome Networks

We next discuss the task of predicting annotations in sparsely characterized genomes. There are currently more than 60 fully-sequenced organisms in the Ensembl repository [1]. The curated annotations of novel model organisms are significantly less than in established systems such as yeast, worm, and fly. A promising direction for the annotation of such organisms is to attempt to exploit the available annotation knowledge about well-studied (*reference*) organisms and transfer it to a novel (*target*) organism at hand. In the presence of interaction networks for both the reference and target organism, one can exploit the interaction information in tandem with the traditional homology-based approaches [21].

In this section, we introduce three hypothesis, originally outlined in [7], that propose a combination of homology and interaction information for robust prediction in sparse genomes. The hypotheses are illustrated schematically in Fig. 3.7a and discussed and evaluated in the remainder of this section.

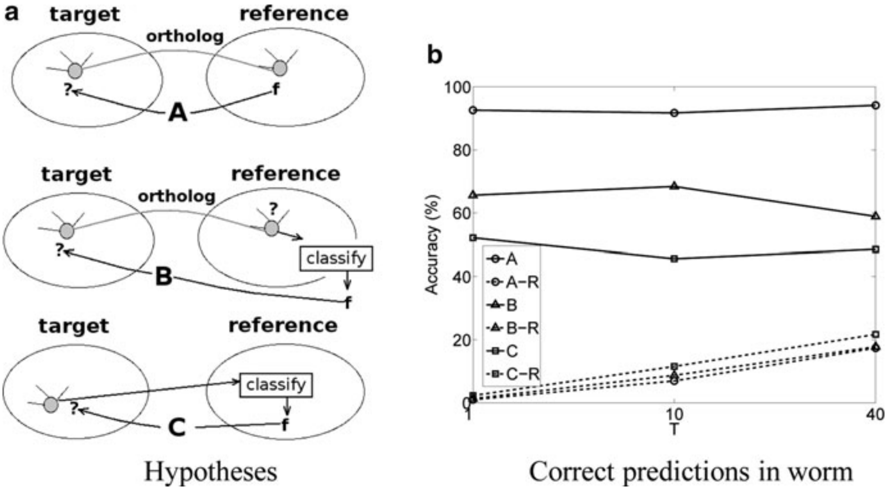


Fig. 3.7 (**a**) Hypotheses A – use of homology; B – prediction + homology; and C – prediction based on ortholog's profile. (**b**) Percentage of correct predictions for reference – yeast and target – worm

6.1 Hypothesis A: Orthologs Share Annotations

If the gene sequences constitute the only information about the target and reference organisms, then based on *Hypothesis A*, a target sequence can be annotated by directly transferring the functions of orthologous sequences from one or more well-studied reference genomes. This approach of comparative annotation is widely adopted and one recent representative study in this category by Hawkins et al. [21] targets prediction of annotations from the GO hierarchy.

6.2 Hypothesis B: Orthologs Share Annotations and Annotation Profiles

For cases in which there is an ortholog mapping but both the target and reference sequences are missing annotation, one can predict the annotation in the reference and transfer it along the ortholog link. Note that *Hypothesis B* allows for increased utility of homologous relationships when the annotations are missing on both ends.

6.3 Hypothesis C: Similar Functional Profiles Imply Shared Annotations

According to *Hypothesis C*, annotations can be transfered between sequences across organisms when their annotation neighborhoods are similar. Note that this hypothesis does not require a homology link to exist between the target and reference sequences.

The utility of the above ideas are examined in [7] using two genomes: yeast (reference) and *C. elegans* (target). The high confidence yeast network *FYI* is used as a reference genome and an interaction network of *C. elegans*, based purely on coexpression analysis [25], is used as a target genome. Worm's genes of single annotation are used as ground truth, and homology information is employed using InParanoid [40]. The results are presented in Fig. 3.7b. The trace annotated *A* in Fig. 3.7b illustrates the percentage of ortholog pairs that share molecular function annotations. For *Hypothesis B*, the prediction accuracy for all possible ortholog pairs is reported in trace *B* in Fig. 3.7b). Annotations of yeast orthologs are first predicted by the *kNN* approach and then transferred to worm. To test whether similar functions have similar neighborhoods across organisms, profiles computed in worm are classified in FYI (trace *C* in Fig. 3.7b). To further shed light on the utility of the three hypotheses, we estimate the predictive power of counterpart random predictors: *A-R*, *B-R* and *C-R* corresponding to each of them.

Note that there are a number of challenges in performing cross-organism classification. The set of GO terms used for annotating yeast and worm do not overlap completely and this could be due to the worm and yeast communities using different levels of the vast GO ontology as well as due to organism-specific functions. The above challenges result in somewhat pessimistic evaluation of Hypotheses B and C. The significant dominance of their corresponding approaches over random predictors is a promising outcome of this initial assessment of their utility. Additional details and evaluation are available in [7].

7 Conclusion

We discussed the utility of interaction networks for the fundamental problem of systematic genome annotation. Existing network annotation predicting techniques are reviewed and categorized. We separated the building blocks of a generalized network prediction algorithm: network synthesis, network feature extraction, and annotation prediction using classification. A recent method that complies with this generalized methodology was introduced together with a comparison evaluation of diverse methods for the prediction task in a model organism. In addition, we presented recent ideas of combining homology and interaction data for the purposes of automated annotation in newly sequenced and sparse genomes using one or more reference well-studied genomes.

Acknowledgments This material is based upon work supported by the National Science Foundation under Grant No. IIS-0917149.

References

1. Ensembl – on-line genome database. http://www.ensembl.org/.
2. Gene expression omnibus. http://www.ncbi.nlm.nih.gov/geo/.
3. The gene ontology. http://geneonetology.org/.
4. Gene ontology: Tool for the unification of biology. *Nature*, 2000.
5. BioGRID: General repository for interaction datasets. *http://www.thebiogrid.org/*, 2006.
6. V. Arnau, S. Mars, and I. Marin. Iterative clustering analysis of protein interaction data. *Bioinformatics*, 2005.
7. Petko Bogdanov and Ambuj K. Singh. Molecular Function Prediction Using Neighborhood Features. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 7(2), 2010.
8. Christine Brun, Francois Chevenet, David Martin, Jerome Wojcik, Alain Guenoche, and Bernard Jacq. Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. *Genome Biology*, 5:R6, 2003.
9. T. Can, O. Camoglu, and A. K. Singh. Analysis of protein interaction networks using random walks. *Proceedings of the 5th ACM SIGKDD Workshop on Data Mining in Bioinformatics*, 2005.
10. Jin Chen, Wynne Hsu, Mong Li Lee, and See-Kiong Ng. Labeling network motifs in protein interactomes for protein function prediction. *ICDE*, 2007.

11. Kyle C Chipman and Ambuj K Singh. Predicting genetic interactions with random walks on biological networks. *BMC bioinformatics*, 10:17, January 2009.
12. H. Chua, W. Sung, and L. Wong. Exploiting indirect neighbors and topological weight to predict protein function from protein-protein interactions. *Bioinformatics*, 2006.
13. H. Chua, W. Sung, and L. Wong. Using indirect protein interactions for the prediction of gene ontology functions. *BMC Bioinformatics*, 2007.
14. C. M. Deane, ukasz Salwinski, Ioannis Xenarios, and David Eisenberg. Protein Interactions: Two Methods for Assessment of the Reliability of High Throughput Observations. *Molecular & Cellular Proteomics*, 1(5):349–356, April 2002.
15. M. Deng, Z. Tu, F. Sun, and T. Chen. Mapping Gene Ontology to proteins based on protein-protein interaction data. *Bioinformatics*, 20:895–902, Apr 2004.
16. M. Deng, K. Zhang, S. Mehta, T. Chen, and F. Sun. Prediction of protein function using protein-protein interaction data. *J. Comput. Biol.*, 10:947–960, 2003.
17. R. Dunn, F. Dudbridge, and CM. Sanderson. The use of edge-betweenness clustering to investigate the biological function in protein interaction networks. *BMC Bioinformatics*, 2005.
18. J.E. Galagan and et. al. The genome sequence of the filamentous fungus neurospora crassa. *Nature*, 422:859–868, 2003.
19. J. Han, N. Bertin, and T. Hao et Al. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 2004.
20. John L Hartman, Barbara Garvik, and Lee Hartwell. Principles for the Buffering of Genetic Variation. *Science*, 291(9):1001–1004, 2001.
21. T. Hawkins, S. Luban, and D. Kihara. Enhanced automated function prediction using distantly related sequences and contextual association by PFP. *Protein Sci.*, 15:1550–1556, Jun 2006.
22. H. Hishigaki, K. Nakai, T. Ono, A. Tanigami, and T. Takagi. Assessment of prediction accuracy of protein function from protein–protein interaction data. *Yeast*, 2001.
23. TK Jenssen, A Laegreid, J Komorowski, and E Hovig. A literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics*, 28(1):21–28, 2001.
24. Ulas Karaoz, T. M. Murali, Stan Letovsky, Yu Zheng, Chunming Ding, Charles R. Cantor, and Simon Kasif. Whole-genome annotation by using evidence integration in functional-linkage networks. *PNAS*, 101:2888–2893, 2004.
25. S.K. Kim, J. Lund, M. Kiraly, K. Duke, M. Jiang, J.M. Stuart, A. Eizinger, B.N. Wylie, and G.S. Davidson. A gene expression map for Caenorhabditis elegans. *Science*, 293:2087–2092, Sep 2001.
26. O. D. King, R. E. Foulger, S. S. Dwight, J. V. White, and F. P. Roth. Predicting gene function from patterns of annotation. *Genome Res.*, 13:896–904, May 2003.
27. Mustafa Kirac and Gultekin Ozsoyoglu. Protein function prediction based on patterns in biological networks. *Research in Computational Molecular Biology*, pages 197–213, 2008.
28. S. Kohler, S. Bauer, D. Horn, and P. N. Robinson. Walking the interactome for prioritization of candidate disease genes. *Am. J. Hum. Genet.*, 82:949–958, Apr 2008.
29. I. Lee, S. V. Date, A. T. Adai, and E. M. Marcotte. A probabilistic functional network of yeast genes. *Science*, 306:1555–1558, November 2004.
30. I. Lee, B. Lehner, C. Crombie, W. Wong, AG Fraser, and E. Marcotte. A single network comprising the majority of genes accurately predicts the phenotypic effects of gene perturbation in caenorhabditis elegans. *Nature Genetics*, 40(2):181–188, 2008.
31. I. Lee, Z. Li, E. M. Marcotte. An Improved, Bias-Reduced Probabilistic Functional Gene Network of Baker's Yeast, *Saccharomyces cerevisiae*. PLoS ONE, 2(10):e988. doi:10.1371/journal.pone.0000988, 2007.
32. Stanley Letovsky and Simon Kasif. Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, 19:i197–i204, 2003.
33. MJ Thompson D Eisenberg TO Yeates M Pellegrini, EM Marcotte. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc Natl Acad Sci USA*, 96(8):4285–8, 1999.

34. K. Maciag, S.J. Altschuler, M.D. Slack, N.J. Krogan, A. Emili, J.F. Greenblatt, T. Maniatis, and L.F. Wu. Systems-level analysis identify extensive coupling among gene expression machines. *Molecular Systems Biology*, 2006.
35. Kathy Macropol, Tolga Can, and Ambuj Singh. Rrw: repeated random walks on genome-scale protein networks for local cluster discovery. *BMC Bioinformatics*, 10:283, 2009.
36. C. Von Mering, M. Huynen, D. Jaeggi, S. Schmidt, P. Bork, and B. Snel. String: a database of predicted functional associations between proteins. *Nucleic Acids Res.*, 2003.
37. H. W. Mewes, D. Frishman, U. Guldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. kMorgenstern, M. Munsterkotter, S. Rudd, and B. Weil. Mips: a database for genomes and protein sequences. *Nucleic Acids Res.*, 30:31–34, 2002.
38. H.W. Mewes and et. al. Overview of the yeast genome. *Nature*, 387:496–512, 1997.
39. E. Nabieva, K. Jim, A. Agarwal, B. Chazelle, and M. Singh. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, 21: i302–i310, 2005.
40. K.P. O'Brien, M. Remm, and E.L. Sonnhammer. Inparanoid: a comprehensive database of eukaryotic orthologs. *Nucleic Acids Res.*, 33:D476–480, Jan 2005.
41. M. Riley. Functions of the gene products of escherichia coli. *FEMS Microbiol. Rev.*, 57: 862–952, 1993.
42. M. Riley. Multifun, a multifunctional classification scheme for escherichia coli k-12 gene products. *Microb Comp Genomics*, 5:205–22, 2000.
43. A. Ruepp and et. al. The genome sequence of the thermoacidophilic scavenger thermoplasma acidophilum. *Nature*, 407:508–513, 2000.
44. A. Ruepp and et. al. The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Res.*, 32:5539–5545, 2004.
45. M. Salanoubat and et. al. Sequence and analysis of chromosome 3 of the plant arabidopsis thaliana. *Nature*, 408:820–822, 2000.
46. Manoj Pratim Samanta and Shoudan Liang. Predicting protein functions from redundancies in large-scale protein interaction networks. *PNAS*, 100:12579–12583, 2003.
47. B. Schwikowski, P. Uetz, and S. Fields. A network of protein-protein interactions in yeast. *Nature*, 2000.
48. R. Sharan, I. Ulitsky, and R. Shamir. Network-based prediction of protein function. *Molecular Systems Biology*, 2007.
49. V. Spirin and L. Mirny. Protein complexes and functional modules in molecular networks. *PNAS*, 2003.
50. Joshua M Stuart, Eran Segal, Daphne Koller, and Stuart K Kim. A gene-coexpression network for global discovery of conserved genetic modules. *Science (New York, N.Y.)*, 302(5643): 249–55, October 2003.
51. A. H. Tong, G. Lesage, G. D. Bader, H. Ding, H. Xu, X. Xin, J. Young, G. F. Berriz, R. L. Brost, M. Chang, Y. Chen, X. Cheng, G. Chua, H. Friesen, D. S. Goldberg, J. Haynes, C. Humphries, G. He, S. Hussein, L. Ke, N. Krogan, Z. Li, J. N. Levinson, H. Lu, P. Menard, C. Munyana, A. B. Parsons, O. Ryan, R. Tonikian, T. Roberts, A. M. Sdicu, J. Shapiro, B. Sheikh, B. Suter, S. L. Wong, L. V. Zhang, H. Zhu, C. G. Burd, S. Munro, C. Sander, J. Rine, J. Greenblatt, M. Peter, A. Bretscher, G. Bell, F. P. Roth, G. W. Brown, B. Andrews, H. Bussey, and C. Boone. Global mapping of the yeast genetic interaction network. *Science*, 303:808–813, Feb 2004.
52. Oron Vanunu and Roded Sharan. A propagation-based algorithm for inferring gene-disease associations. *German Conference on Bioinformatics*, 2008.
53. Y. Wu and S. Lonardi. A linear-time algorithm for predicting functional annotations from PPI networks. *J Bioinform Comput Biol*, 6:1049–1065, Dec 2008.
54. G. X. Yu, E. M. Glass, N. T. Karonis, and N. Maltsev. Knowledge-based voting algorithm for automated protein functional annotation. *PROTEINS: Structure, Function, and Bioinformatics*, 61:907–917, 2005.
55. Shi-Hua Zhang, Hong-Wei Liu, Xue-Mei Ning, and Xiang-Sun Zhang. A hybrid graph-theoretic method for mining overlapping functional modules in large sparse protein interaction networks. *IJDMB*, 3(1):68–84, 2009.

Chapter 4
Proteome Network Emulating Models

Phuong Dao, Fereydoun Hormozdiari, Iman Hajirasouliha, Martin Ester, and S. Cenk Sahinalp

Abstract The proteome network (or protein–protein interaction (PPI) network) of an organism represents each protein as a vertex and each pairwise interaction as an edge. In the past 10 years, we witnessed a significant amount of effort going into the development of the PPI networks and the computational tools for analyzing them. In particular, there have been several attempts to capture the topological features of PPI networks through random graph models, which have been successfully applied to the emulation of "small-world" networks, which are sparse, but highly connected. The available PPI networks have also been thought to have a small diameter with power-law degree distribution thus "scale-free" network emulators such as the Preferential Attachment Model have been investigated for the purposes of emulating PPI networks. The lack of success in this direction led to the development of further models, which either reject the "scale-freeness" of the PPI networks, such as the Geometric Random Network Model or guarantee scale freeness through means of expansion other than "Preferential Attachment" such as vertex (i.e., protein/gene duplication) – as in the case of the Pastor-Satorras Model or the more recent Generalized Duplication Model. In this study, we compare available PPI networks of various sizes with those generated by the random graph models and observe that the Generalized Duplication Model, with the "right" choice of the initial "seed" network, provides the best alternative in capturing all network feature distributions. One network feature distribution that remains difficult to capture, however, is the "dense graphlet" distribution: all available PPI networks seem to include (many) more dense graphlets such as cliques in comparison to the networks generated by all available models.

P. Dao (✉)
School of Computing Science, Simon Fraser University, 8888 University Drive,
Burnaby, BC, Canada V5A 1S6
e-mail: pdao@cs.sfu.ca

M. Koyutürk et al. (eds.), *Functional Coherence of Molecular Networks in Bioinformatics*, DOI 10.1007/978-1-4614-0320-3_4,
© Springer Science+Business Media, LLC 2012

1 Proteome Network Emulation Models

The proteome network of an organism is typically modeled by a graph in which each node represents a protein and each edge represents an interaction between a pair of proteins. Significant amount of effort has been put toward developing a better understanding of proteome network topology, which together with an improved understanding of its dynamics would provide a deeper insight into the inner working of a cell. The study of proteome network topology is also significant in understanding the mechanisms through which known drugs work against complex diseases and developing novel therapies.

The theoretical study of proteome network topology started about 10 years ago with the now well-known observations on the structure of the Yeast proteome network obtained through 2-hybrid assays [18, 37]:

1. The degree distribution of nodes (i.e., the proportion of nodes with degree k as a function of k) approximates a *power-law* (i.e., is approximately ck^{-b} for some constants c, b) – this implies that the expected maximum degree in the graph is proportional to $n^{1/b}$.
2. The graph exhibits the *small world effect*: the shortest distance between a randomly selected pair of nodes is "small."

Small world phenomena and the power-law degree distribution with high maximum degree have previously been observed in a number of naturally occurring graphs such as communication networks [12], web graphs [1, 3, 7, 9, 19, 20], research citation networks [30], human language graphs [13], neural nets [39], etc. These two properties cannot be observed in the classical random graph models studied by Erdös and Rényi [31] in which edges between pairs of nodes are determined independently. It is well known that on such a graph with fixed number of vertices, n, between which the edges between a pair of vertices are placed uniformly at random with probability $p = \text{avgdegree}/n$, the maximum degree is expected to be logarithmically related to the number of vertices. On the other hand, alternative random graph models, as will be described below, do have power-law degree distributions and high maximum degree [3, 7, 38] – thus it has been tempting to investigate whether these models agree with other topological features of the proteome networks.

There are two well-known models that provide power-law degree distributions (see [4, 8, 9]). The *Preferential Attachment* Model [1, 7], was introduced to emulate the growth of naturally occurring networks such as the web graph; unfortunately, it is not biologically well-motivated for modeling proteome networks. The *Duplication Model*, on the other hand [6, 26, 36], is inspired by Ohno's hypothesis on genome growth by duplication [24]. Both models are iterative in the sense that they start with a *seed graph* and grow the network in a sequence of steps.

Another model that has been studied for the purposes of protein–protein interaction (PPI) network emulation is the Geometric Random Graph Model [10, 21, 23]. The Geometric Random Graph Model is inspired by the fact that a protein can be described through a collection biochemical properties which can be represented by a real-valued vector. Thus, proteins could be considered as points in some ℓ-dimensional Euclidean space. The Geometric Random Graph Model thus picks n random points on some low dimensional Euclidean space (2-D or 3-D are most common) each of which represents a vertex. Two vertices are then connected if their distance is less than some given threshold r. The Geometric Random Graph Model neither satisfy the small world property, nor provide a power-law degree distribution.

In what follows, we describe each of the above models in more detail. Later, we discuss how well these models emulate available PPI networks with respect to capturing their topological features.

1.1 Preferential Attachment Models

The Preferential Attachment Model dates back to Yule [41] and Simon [32]. It was later reintroduced for purposes of modeling the world wide web by Barabási and Albert [3], and was investigated more formally by Bollobás and Riordan [7]. As mentioned above, the Preferential Attachment Model is an iterative model which generates exactly one vertex per iteration. Let $G(t) = (V(t), E(t))$ be the graph at the end of time step t, where $V(t)$ is the set of nodes and $E(t)$ is the set of edges/connections. Let v_t be the node generated in time step t. In step t, the Preferential Attachment Model generates v_t and connects it to every other node v_τ independently with probability $c \cdot d_{t-1}(v_\tau)/2|E(t-1)|$, where c is the average degree of a node in G; that is, v_t prefers to connect itself to high degree nodes.

Bollobás and Riordan [7] showed that with high probability the diameter of a graph constructed in this way was $\sim \log t/\log\log t$; here, t stands for the time step and thus (is approximately) the number of nodes. Subsequently, Bollobás et al. [7] proved that the degree sequence of these graphs follow a power-law distribution. Attention has also been given to models where the attractiveness of vertices fades over time, for example [34]. More recently, Cooper and Frieze [9] gave a general analysis of random graph processes revealing that many graphs generated by Preferential Attachment exhibit power-law degree distributions. This analysis, and those of [20, 29, 33]; obtained graphs with a power-law parameter larger than 2 but smaller than 3 by using a graph generation model that allows edge insertion between existing nodes.

1.2 Geometric Random Model

As mentioned earlier, representation of proteins with real-valued vectors (thus points in Euclidean space) is suitable for certain applications. The Geometric Random Graph Model [10, 21, 23] thus tries to build a random graph $G = (V, E)$ in an ℓ-dimensional Euclidean space (ℓ is typically $2, 3$ or 4). The model independently

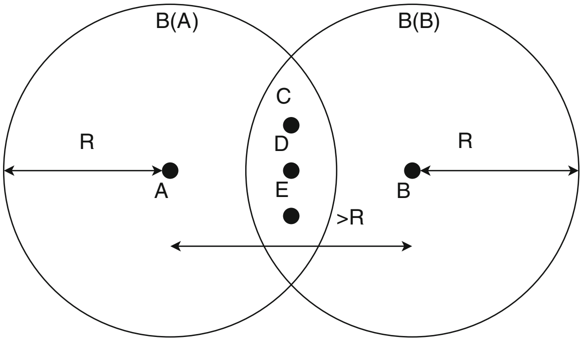


Fig. 4.1 $K_{3,2}$ cannot exist in any two dimensional GRG: if A and B are in the same partition, their distance must be at least r. Because C, D, E are in the same partition, each must be within distance r to both A and B – thus they must be at the intersection of the balls centered in A and B, each with radius r. However, that implies that the pairwise distances between C, D, and E cannot be all greater than r – which implies a contradiction

picks some n points in the Euclidean space uniformly at random and assigns a vertex to each such point. Two vertices are then connected if their (Euclidean) distance is below some threshold value r. The value of r can be picked in a way that the resulting graph ends up with the desired average degree, that is, that of the emulated network [10] – on an ℓ dimensional, unit Euclidean cube, r is in the order of $(\text{avgdegree}/n)^{1/\ell}$.

One of the limitations of the Geometric Random Model is that the resulting graphs cannot contain certain subgraphs commonly observed in PPI networks such as the complete bipartite graph ($K_{3,2}$ for $\ell = 2$ and $K_{3,3}$ for $\ell = 3$) [28]:

For $\ell = 2$, for example, consider a $K_{3,2}$, where A, B are points of the "left" partition and C, D, and E are points of the "right" partition. For a given point P, let $\beta(P)$ be the "ball" centered at P with predefined radius r. Points C, D, and E should be at the intersection of two balls $\beta(A)$ and $\beta(B)$. However, this is impossible since their distances must be greater than r as illustrated in Fig. 4.1.

It is interesting that in [28], it was conjectured that a geometric random graph generated in ℓ-dimensional space cannot contain a complete bipartite graph, $K_{\ell+1,2}$. This conjecture was later disproved (Oliver King, personal communication). In fact $K_{4,2}$ can be embedded to a geometric graph with 3 dimensions by embedding the bigger partition on a square (each side length r) and the smaller partition on the two sides of a line crossing the center of the square orthogonally, with distance r to each other. However, emulations show that $K_{4,2}$ (and other complete bipartite graphs) can be produced by random geometric models very rarely.

Note that although a *Random* Geometric Model may not accurately capture the topological features of a PPI network, it is possible to come up with a *deterministic* geometric network – one whose vertices in the Euclidean space is derived from the PPI network itself – that is highly similar to the PPI network in question. Such an "embedding" of the PPI network to a low dimensional Euclidean space can be obtained as follows. Given a PPI network $G = (V,E)$, let $D_{i,j}$ be the length of the shortest path between a pair of vertices i and j, where the distance between vertices connected through an edge is 1. Clearly, $D_{i,j}$ defines a metric. The embedding is a bijection from vertices in the PPI network and vectors $x_1,\ldots,x_n$ in R^m for a given m $(= 2,3$, or 4) such that $|x_i - x_j| \approx D_{i,j}$ for all i,j – thus the closer the two vertices are in the PPI network, the closer they will be on the geometric network.

Since $D_{i,j}$ satisfies the triangle inequality, doubling and then centering the matrix $D_{i,j}$ as per below produces a symmetric and positive semi-definite matrix A:

$$A_{ij} = -\frac{1}{2}\left(D_{ij}^2 - \frac{1}{n}\sum_k D_{ik}^2 - \frac{1}{n}\sum_k D_{kj}^2 + \frac{1}{n^2}\sum_k\sum_l D_{kl}^2\right)$$

Let $\lambda_1,\ldots,\lambda_t$ be the t positive eigenvalues of A ordered from the largest to the smallest and $u_1,\ldots,u_t$ be the corresponding eigenvectors. Let $X \in \mathbb{R}^{t\times n}$ be a matrix such that $X^T X = A$ and $X_1,\ldots,X_n \in \mathbb{R}^t$ be the columns of X, it can be shown that $||X_i - X_j||_2 = D_{ij}$. Ideally, when $m = t$, the set of columns $X_1,\ldots,X_n$ is the set of points $x_1,\ldots,x_n$ in $\mathbb{R}^m$ that one would like to find:

$$X = \begin{bmatrix} \sqrt{\lambda_1} u_1^T \\ \vdots \\ \sqrt{\lambda_t} u_t^T \end{bmatrix}$$

In the case that $m \leq t$, we find the matrix $\widehat{X}$ with rank m (X has rank t) such that it is the closest matrix to X. More specifically, it is possible to minimize the matrix norm (Frobenius norm) of $X - \widehat{X}$

$$\left(\sum_i\sum_j (X_{ij} - \widehat{X}_{ij})^2\right)^{1/2}$$

and $\widehat{X}$ could be obtained by picking the m eigenvectors corresponding to m largest eigenvalues of A

$$\widehat{X} = \begin{bmatrix} \sqrt{\lambda_1} u_1^T \\ \vdots \\ \sqrt{\lambda_m} u_m^T \end{bmatrix}$$

1.3 Gene Duplication-Based Models

In a number of naturally occurring graphs with power-law degree distributions such as proteome networks, peer to peer networks, and in a limited way the web graph, the mechanism underlying the growth process seems to be different from that in Preferential Attachment models. Rather, these networks seem to grow via node duplications. For example, Ohno's theory of genome evolution [24] states that the main driving force behind genome and thus proteome growth is gene duplication followed by point mutations. In peer to peer networks, a new user typically chooses the servers used by an existing node; similarly, new web pages tend to share content with related web pages. Possibly, the first analysis of a duplication-based random graph model is given in [20]; this model generates directed graphs with constant outdegree. A more general version of this directed model was later analyzed in [9]. In both of these models, the (out)degree of the newly generated node is bounded by a constant and does not depend on the degree of the duplicated node.

A less constrained duplication-based random graph model where the degree of the newly generated node is not bounded by a constant was introduced in [6, 26, 36]. In this model, at each iteration t, one existing node is chosen uniformly at random and is "duplicated" with all its edges. Then, in a "divergence" move, (1) each existing edge of the new node is deleted with probability q and (2) a new edge is generated between the new node and every other node with probability r/t. This last step is referred to as "mutation" by some authors (e.g., [17]).

More specifically, the duplication model above, which we will denote as the "Pastor-Satorras" Model, starts with an arbitrary connected network $G(t_0)$, of size t_0. For $t > t_0$, let $G(t-1)$ be the network at the end of time step $t-1$. At iteration t, exactly one new node, denoted as v_t, is added to $G(t-1)$ as follows:

A node w is picked uniformly at random from $G(t-1)$, and w is "duplicated" to create the new node v_t which is initially connected to all the neighbors $N_{t-1}(w)$ of w, but not to w itself. The edges initially incident to v_t are then updated in the following way:

Step 1. Duplication: Each edge $e = (v_t, u)$, $u \in N_{t-1}(w)$ is independently deleted with probability q or retained with probability $p = 1 - q$.

Step 2. Uniformly at random edge addition: Each node u of $G(t-1)$ independently connected to v_t with probability $r/(t-1)$, where r is a nonnegative constant of the process, and any parallel edges created are merged.

The first analysis of the Pastor-Satorras Model [26] suggested that the degree distribution of the resulting network is a "power-law with exponential cut-off." This means that f_k, the fraction of nodes with degree k among all nodes, is independent of time and is approximated by $f_k = ck^{-b} \cdot a^{-k}$; here a, b, c are constants. However, the analysis in [26] makes a number of simplifying assumptions to get this result. For instance, it approximates the probability of generating a node with degree k by the probability of duplicating a node with degree $k+1$ only and subsequently deleting one of its edges.

A later analysis of the Pastor-Satorras Model, by Chung et al. [8] for the special case that $r = 0$ – which will be referred to as the "Pure Duplication Model" suggests that the asymptotic tendency for the degree distribution is a standard power-law – rather than one with a cut-off. More specifically, Chung et al. suggested that the fraction of nodes with degree k should asymptotically approach a power-law distribution of the form $f_k = ck^{-b}$, where the value of b can be derived from the equation $1 = bp - p + p^{b-1}$; note that it is possible to have $b \leq 2$ for some values of q.

A final analysis of the Pastor-Satorras Model by Bebek et al. [4] reveals that:

(1) the Pastor-Satorras Model in general produces many *singletons*, that is, nodes which are not connected to any other node;
(2) in the Pure Duplication Model ($r = 0$), in particular when $p = 1/2$, all but a logarithmic number of nodes will end up as singletons – in other words the proportion of singletons to all nodes in the network asymptotically approaches 1;
(3) for different values of p and r, the number of singletons generated by the Pastor-Satorras Model is much higher than number of singletons in known PPI networks.

As a result of the above observations, the Pure Duplication Model should asymptotically satisfy $f_0 = 1$ and $f_k = 0$, for all $k > 0$. For the case $q = 0.5$, the average degree of nodes in the Pure Duplication Model does not change over time. Thus (1) the average degree of nonsingletons must increase in time and (2) there is a single connected component of size $o(t)$ with increasing average degree. It is possible that this connected component of the network generated by the Pure Duplication Model exhibits a power-law with parameter $b \leq 2$; unfortunately, this is difficult to establish analytically [4].

The above observations reveal that the Pastor-Satorras Model is not suitable for emulating networks of interest and thus several modifications to the Pastor-Satorras Model have been suggested [4, 5, 16]. Each of these models differ from the Pastor-Satorras Model by having an extra edge generation step in each iteration. Note that it is desirable for such a step to not only maintain that $G(t)$ stays as a single connected component, but also to restrict the number of the edges added to the network during the uniformly at random step.

Version 1: **Step 3.** If v_t has become a singleton at the end of the duplication move, it is connected to $a_1 \geq 1$ uniformly chosen random nodes.

Version 2: **Step 3.** The node v_t is connected to $a_2 \geq 1$ additional nodes chosen uniformly at random. This occurs even if v_t has not become a singleton at the end of the duplication move.

Version 3: **Step 3.** If v_t has become a singleton at the end of the duplication move, it is deleted.

Bebek et al. show that in each of these versions of the "Generalized Duplication Model," (1) no singletons are generated, (2) the network stays connected, and (3) the degree distribution of the nodes exhibit a power-law of the form $f_k = ck^{-b}$. Versions $i = 1, 2$ also ensure that the minimum degree of a node would be a_i. However, Version 3 is more powerful in the sense that the parameters of the model can be picked in a way to ensure that the expected degree of a node stays as a user defined constant [4] as will be demonstrated below.

1.3.1 Parameter Selection in the Generalized Duplication Model

The deletion probability $1-p$ and the insertion probability r in Version 3 of the Generalized Duplication Model can be chosen so as to maintain the expected (over all executions of the model) average (over all nodes) degree of the network as a user defined constant throughout the iterations. Let $G(t) = (V(t), E(t))$ be the network generated by the Modified Duplication Model and let $n(t) = |V(t)|$ and $e(t) = |E(t)|$. Also, let $n_k(t)$ be the number of nodes in time step t with degree k and $a(t)$ be the average degree of nodes in $G(t)$. Finally, let $P_k(t) = n_k(t)/n(t)$, the frequency of nodes with degree k at time step t. We assume that $P_t(k)$ is asymptotically stable, that is, $P_k(t) = P_k(t+1)$ for all $1 \le k \le t$ for sufficiently large values of t. In other words, we assume that $P_k(t) = d_k$ for some fixed d_k. By definition

$$a(t) = \sum_{k=1}^{t} k \cdot \frac{n_k(t)}{n(t)} = \sum_{k=1}^{t} k \cdot P_k(t) = \sum_{k=1}^{t} k \cdot d_k.$$

Now, we can calculate the average degree $a(t+1)$ under the condition that degree frequency distribution is stable and $a(t) = a$, a constant.

$$\text{Exp}[e(t+1)] = e(t) + \sum_{k=1}^{t} k \cdot P_k(t) \cdot p + r = \frac{n(t) \cdot a(t)}{2} + p \cdot a(t) + r.$$

Let $Pr_s(t)$ be the probability that v_{t+1} ends up as a singleton.

$$Pr_s(t) = \sum_{k=1}^{t} P_k(t) \cdot (1-p)^k \cdot \left(1 - \frac{r}{n(t)}\right)^{n(t)-k} \approx \sum_{k=1}^{t} d_k \cdot (1-p)^k \cdot \frac{1}{e^r}.$$

Since this probability does not depend on t asymptotically, we can set $Pr_s(t) = Pr_s$. Now, we can calculate the expected number of nodes and the expected number of edges in step $t+1$.

$$\begin{aligned}
\text{Exp}[n(t+1)] &= Pr_s \cdot n(t) + (1 - Pr_s) \cdot (n(t)+1). \\
\text{Exp}[e(t+1)] &= \text{Exp}\left[\frac{n(t+1) \cdot a(t+1)}{2}\right] = \frac{a}{2} \cdot \text{Exp}[n(t+1)] \\
\text{Exp}[e(t+1)] &= \frac{a}{2} \cdot (Pr_s \cdot n(t) + (1 - Pr_s) \cdot (n(t)+1)).
\end{aligned}$$

Comparing the above equation with the first equation for $\text{Exp}[e(t+1)]$, we get

$$\frac{a}{2} \cdot (Pr_s \cdot n(t) + (1 - Pr_s) \cdot (n(t)+1)) = \frac{n(t) \cdot a(t)}{2} + p \cdot a(t) + r = \frac{n(t) \cdot a}{2} + p \cdot a + r.$$

Solving the above equation results in $a = 2r/(1 - Pr_s - 2p)$, where Pr_s is a function of p, r, and d_k only.

The discussion above demonstrates that the two key parameters p and r of the Generalized Duplication Model are determined by the degree distribution (more specifically the slope of the degree distribution in the log–log scale) and the average degree of the PPI network we would like to emulate. Perhaps due to the strong evidence that the seed network does not have any effect on the asymptotic degree distribution [5], the role of the seed network (the only free parameter remaining) in determining other topological features of the Duplication Model has not been investigated.

2 Assessing Network Evolution Models

There are several topological measures that can be used to test whether two networks (e.g., a natural and an emulated network) are similar or not, starting from rigorous measures such as Approximate Graph Isomorphism, to relaxed "measures" based on topological characteristics, such as the degree distribution.

2.1 Graph Isomorphism

Two networks $G(V,E)$ and $G'(V',E')$ are called *isomorphic* if there exists a bijective mapping f from the vertex set of G to the vertex set of G', such that two vertices v and w are connected in G if and only if $f(v)$ and $f(w)$ are connected in G'; the Graph Isomorphism problem thus asks to computationally verify whether two graphs are isomorphic. Similarly, the subgraph isomorphism problem asks whether graph G is isomorphic to a subgraph of G'. The Graph Isomorphism problem has received a lot of attention from the theoretical computer science community partially because it is one of the very small number of problems which are neither known to be polynomial time solvable, nor are NP-complete [14]. On the other hand, subgraph isomorphism is a well-known NP-complete problem [14].

A more general version of the Graph Isomorphism problem is the Approximate Graph Isomorphism problem, which can be defined as the minimum number of (1) vertex deletions (together with all edges incident on a vertex), or alternatively, (2) edge deletions as well as singleton (a vertex with no connections) deletions from the two input graphs so that the resulting graphs are isomorphic. Approximate Graph Isomorphism as defined above provides the ultimate measure for PPI network similarity, however, because its polynomial time solution implies a polynomial time solution to the subgraph isomorphism problem (for both variants), it is NP-hard.

2.2 Network Features

More relaxed means to assess the similarity of two networks may be based on common topological features which are (1) easy to detect/quantify (it should not take "too much time" to detect/count these features) and (2) reasonably robust (i.e., minor changes in the network topology should not change these features substantially). We provide examples for such features below.

2.2.1 Degree Distribution

The first and foremost topological feature used to compare graphs (PPI networks, communication networks, web graphs, or graphs generated by random models) is the *degree distribution*, that is, the distribution of the number of nodes with specific number of connections. Some natural networks such as the Internet physical connection network, the web graph or many PPI networks all seem to have a degree distribution in the form of a power-law [18, 37], that is, the frequency of nodes with degree k is approximately ck^{-b} for some constants c, b. This is in contrast with the well-known Erdos–Renyi random graphs where each edge (between a pair of vertices) is established randomly, independent from the remainder of the graph. In the log–log scale, the power-law characteristics of a degree distribution is easy to observe as it provides a straight line – compare this with the Erdos–Renyi graphs where the frequency of the nodes with degree k is logarithmic with $1/k$.

Note that the power-law nature of the degree distribution in some of these networks have been disputed and the power-law like behavior has been attributed to sampling issues, experimental errors, or statistical mistakes [11, 15, 21, 27, 35].

2.2.2 *k*-hop Reachability

Let $V(i)$ denote the set of nodes in V whose degree is i. Given a node v, let its k-hop degree, that is, the number of distinct nodes it can reach in at most k hops be denoted by $d(v,k)$. We define $f(i,k)$, the k-hop reachability of $V(i)$ as

$$f(i,k) = \frac{1}{|V(i)|} \sum_{w \in V, d(w)=i} d(w,k)$$

Thus, $f(i,k)$ is the "average" number of distinct nodes a node with degree i can reach in k hops; for example, $f(i,1) = i$ by definition. As per the degree distribution, the k-hop reachability provides a means for comparing the "connectivity" of two networks.

2.2.3 Betweenness Distribution

The betweenness of a given node of a network measures the extent to which this node lies "between" node pairs in the network $G = (V,E)$. The formal definition of betweenness is as follows. Let $\sigma_{x,y}$ be the number of shortest paths from $x \in V$ to

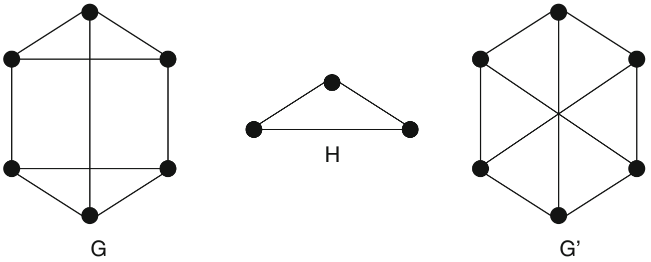


Fig. 4.2 The two networks G and G', each with six vertices, have identical degree and closeness distributions. However, G includes two distinct subnetworks isomorphic to the triangle subgraph H, whereas G' includes none

$y \in V$ for all pairs of $x, y \in V$ (note that in undirected graphs $\sigma_{x,y} = \sigma_{y,x}$). Let $\sigma_{x,y}(v)$ be the number of shortest paths from $x \in V$ to $y \in V$ which goes through node v. The betweenness $\mathrm{Bet}(v)$ of node v is thus defined as

$$\mathrm{Bet}(v) = \sum_{(i,j) \in V, i,j \neq v} \frac{\sigma_{i,j}(v)}{\sigma_{i,j}}$$

2.2.4 Closeness Distribution

For all $x, y \in V$, we define $d_{x,y}$ as the length of the shortest path between x and y. The closeness of a node $v \in V$ is defined as

$$\mathrm{Cls}(v) = \frac{|V| - 1}{\sum_{i \in V} d_{v,i}}.$$

2.2.5 Subgraph Frequency

The normalized number of occurrences of particular subgraphs in a given network can provide alternative means of measuring similarity among networks. In Fig. 4.2, for example, two networks G and G' are far from being isomorphic; however, they have not only the same degree distribution (there are only degree 3 vertices in each) but also the same closeness distribution. Nonetheless, the number of particular subgraphs in G and G' are entirely different: for example, the triangle subgraph (depicted as H) does not appear in G' but occurs twice in G.

More specifically, a *graphlet* is a small connected and induced subgraph of a large graph such as a triangle or a clique. The *graphlet count* of a given graphlet g with r nodes in a given graph $G = (V, E)$ is defined as the number of distinct subsets of V (with r nodes each) whose induced subgraphs in G are isomorphic to g. Note that an induced subgraph (more accurately, a vertex induced subgraph) of a network

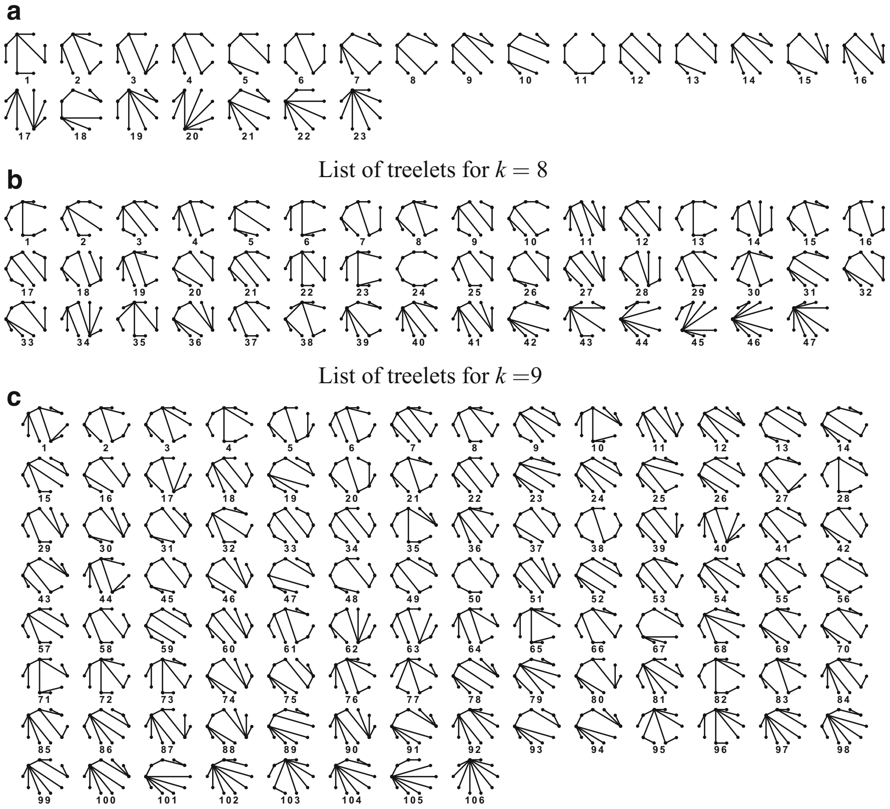
G is a subset of the vertices of G together with all edges whose endpoints are both in this subset; that is, G' is an induced subgraph of G if and only if for each pair of vertices v' and w' in G' and their corresponding vertices v and w in G, either there are edges between both v', w' pair and v, w pair or there are no edges between any of the pairs. If a subgraph G' of G is not an induced subgraph (i.e., it does not include all the edges in G that are present between node pairs in G'), then it is called a noninduced subgraph of G.

Note that the problem of detecting an induced occurrence of an arbitrary (i.e., arbitrarily complex) subgraph in a network is NP-Complete (e.g., detecting the presence of a clique of a given size is NP-Complete). However, given a graph with n nodes, it is possible to not only detect the presence but also get an accurate count of noninduced subgraphs especially those which are trees or have bounded treewidth and have $O(\log n)$ vertices, in time polynomial with n [22, 25]; there are also heuristics for counting subgraphs with "high" density [28] that work well especially in sparse networks, including the PPI networks.

In a given graph $G(V,E)$, the *(treelet count)* of a given tree T with k nodes is defined as the number of distinct subsets of V (with k nodes) whose noninduced subtrees in G are isomorphic to T. The motivation for considering noninduced subtrees is that available PPI networks are far from complete and error free; the interactions between proteins reported by these networks include both false positives and false negatives. Thus, an occurrence of a specific subtree in one network may include additional edges in its occurrence in another network and vice versa. It is possible to get the treelet distribution of all tree topologies with $O(\log n)$ vertices via the color coding-based algorithm of Alon et al. [22, 25]; this technique is also applicable to subgraphs with bounded treewidth. Let the number of distinct noninduced occurrences of treelet T of k vertices, in network G be denoted as $n(T,G)$. Given a fixed approximation factor ε the algorithm by Alon et al. returns in $2^{O(k)}n^{O(1)}$ time, a $n'(T,G)$ such that $(1-\varepsilon)n(T,G) \le n'(T,G) \le (1+\varepsilon)n(T,G)$; note that this is a randomized algorithm with success probability δ – and the constants in the running time are implicitly dependent on the value of δ. Clearly for $k = O(\log n)$, the algorithm has a polynomial running time.

Given a network G and a graphlet g, we denote by $o(g,G)$ the number of induced occurrences of g in G and by $o'(g,G)$ the number of noninduced occurrences of g in G. Note that the typical graphlets that are considered for noninduced occurrence distributions are trees. For a list of graphlets $L = \{g_1, \ldots, g_k\}$, the vector $graphlet(L,G) = [o(g_1,G), \ldots, o(g_k,G)]$ denotes the "induced graphlet distribution" of G; similarly, the vector $graphlet'(L,G) = [o'(g_1,G), \ldots, o'(g_k,G)]$ denotes the "noninduced graphlet distribution" of G. In case we are given a graphlet list $L' = \{t_1, \ldots, t_k\}$, which includes all possible trees of a particular size, the vector $treelet(L',G) = [f(t_1,G), \ldots, f(t_k,G)]$ denotes the "normalized noninduced treelet distribution" of G where $f(t_i,G) = o'(t_l,G)/\sum_{l=1}^{k} o'(t_l,G)$.

There are 141 possible unique graphlets/subgraph topologies with 3, 4, 5, and 6 nodes, as shown in Fig. 4.3. Furthermore, there are 106 possible unlabeled and unordered tree topologies with ten vertices, an additional 47 trees with nine vertices, and 23 trees with eight vertices, as shown in Fig. 4.4. In what follows, we will focus



List of treelets for $k = 8$

List of treelets for $k = 9$

List of treelets for $k = 10$

Fig. 4.3 List of all graphlets with 3,4,5, and 6 vertices and cliques with 7–10 vertices

on graphlet distributions for all induced graphlet topologies with 3,4,5, and 6 nodes as well as distributions of cliques with 7, 8, 9, and 10 nodes. Later, we provide results on normalized, noninduced treelet distributions with 8,9, and 10 nodes.

2.2.6 Robustness of Network Features

As explained earlier, it is of key importance for a chosen network feature to be resilient to "minor" perturbations in the network for the purposes of network comparison. On the other hand, the chosen feature should be fairly sensitive to "major" changes in the network topology. To analyze the robustness of the network features summarized earlier, we have to have a model for perturbating the network. Here, we will focus on a simple model that alters the edges in a random i.i.d. fashion – without changing the number of edges or vertices. More specifically,

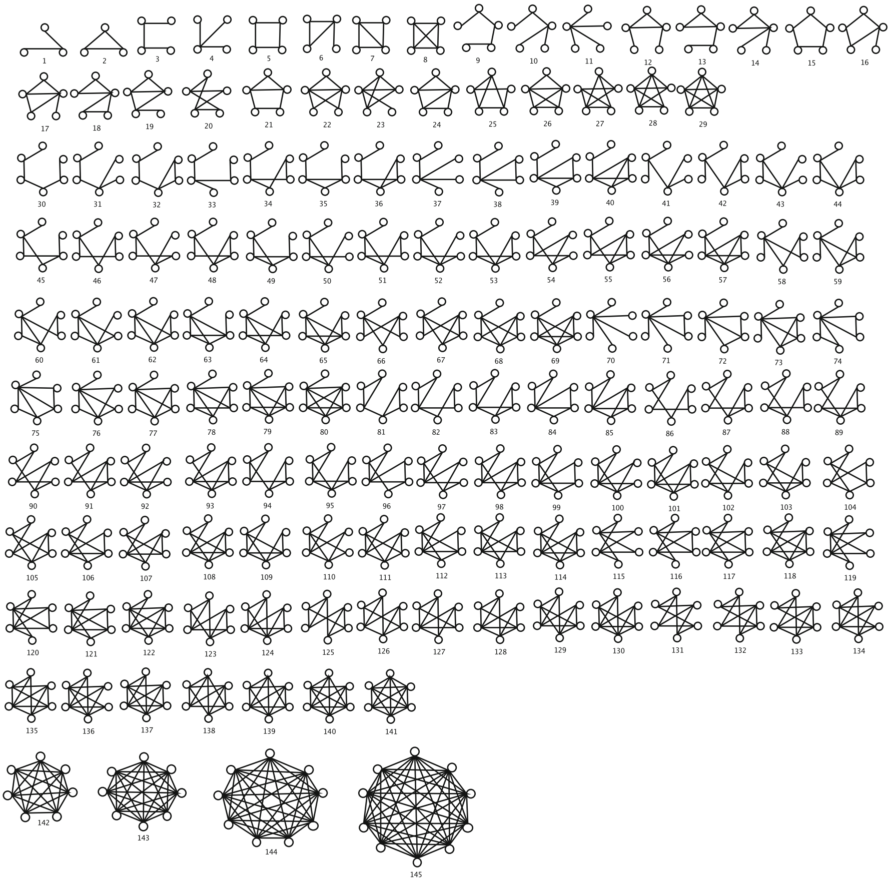


Fig. 4.4 List of all treelets with $k = 8, \ldots, 10$ vertices

the model deletes each edge, independently, with some fixed probability p and for each deleted edge, places another one between a pair of vertices (which are not already connected), chosen uniformly at random. To maintain connectivity, the model treats the deletion of edges which are incident to a vertex with degree 1 (in a connected network, an edge can be incident to at most one vertex with degree 1): if such an edge is chosen for deletion, then the corresponding edge insertion will be between the original vertex with degree 1 and another randomly chosen vertex.

The above network perturbation model allows us to quantify the proportional perturbation in the network. We compare k-hop, betweenness, closeness, and graphlet distributions of networks generated by random graph models with varying levels of perturbations and the Yeast PPI network. For k-hop, betweenness, and closeness values, we plot for each possible value x, the number of nodes whose corresponding feature value is at least x. For the graphlets, we simply rank them with respect to robustness (higher indices imply lower robustness) and plot out the number of induced or noninduced occurrences of each graphlet corresponding to each index value.

As can been seen from Fig. 4.5, all the network features introduced earlier are fairly robust. More specifically, when a minor proportion (i.e., 10–20%) of edges are perturbed the "distribution" of the feature does not change significantly: see Fig. 4.5 where the red plot represent the Yeast PPI network features, while green plot represents the same network with a proportional perturbation of 10% and the blue plot represents the network with a proportional perturbation of 20%. Furthermore, the network features seem to be sensitive to considerable proportional perturbations (i.e., 50% or more) as can be seen in Fig. 4.5; here, the red plot represents the Yeast PPI network, the yellow plot represents 50% proportional perturbation, and the black plot represents 60% proportional perturbation. As can be seen in this figure, the distribution of all network features described earlier smoothly change as the perturbation on the original PPI network increases. In fact, the higher the proportion of the perturbations, the more "Erdos–Renyi like" the network becomes – as will be demonstrated in the next section. Although this does not mean that each network feature equally captures each alteration in the network equally well, we can expect some changes in the distribution of each feature under significant alterations to the network.

2.3 Emulating PPI Networks

As mentioned earlier, various random network generation models have been devised to emulate natural PPI networks of the Yeast, *E.coli*, *H.pylori*, and other organisms. Table 4.1 shows the number of vertices and edges of the PPI networks used in our study from *Database of Interacting Proteins* [44]. Here, we consider only the largest connected components of Yeast, *H.pylori*, and *E.coli* PPI networks. The models considered here are the Erdös–Rényi Model, the Preferential Attachment Model, the Random Geometric Model, and the Generalized Duplication Model (the version which deletes a singleton as soon as it is generated).

We first compare five independent networks generated by the Erdös–Rényi Model against the Yeast PPI network. As can be seen in Fig. 4.6, the Erdös–Rényi Model does not emulate the Yeast PPI network well.

When the Yeast PPI network is compared against the Preferential Attachment Model, the results are significantly better: the average degree of the Yeast PPI network is 7, hence, we picked the Preferential Attachment parameters so as to obtain an average degree of 7 – this simply requires that the value of c is 7. See Fig. 4.7 for a comparison of five independently generated networks using the Preferential Attachment Model against the Yeast PPI network.

The Random Geometric Model in 4-D Euclidean space provides significantly poorer results when emulating the Yeast PPI network. Here, the parameter r is

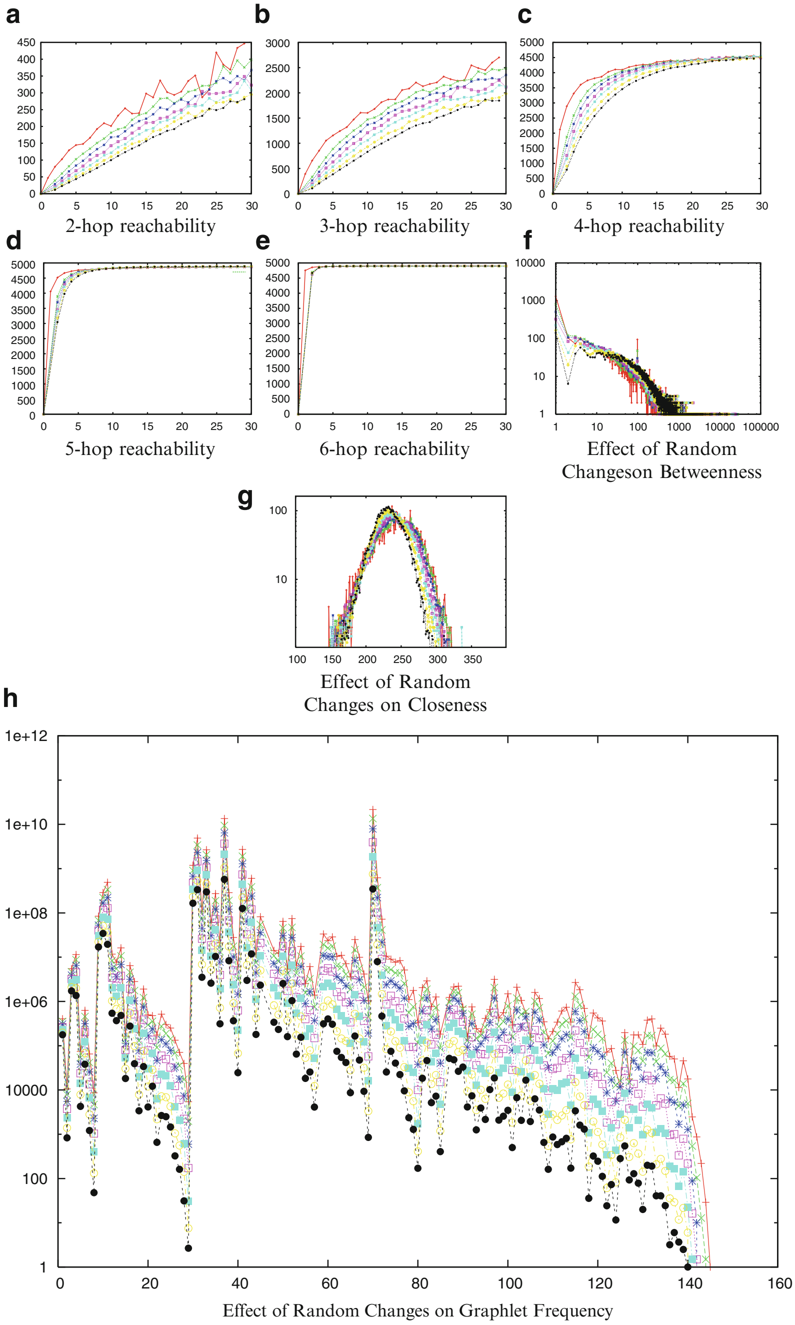


Fig. 4.5 The effect of random changes on the *k*-hop reachability, betweenness, closeness distributions, and graphlet frequency of yeast (*red*), 10% change in edges of yeast PPI network (*green*), 20% change (*blue*), 30% change (*purple*), 40% change (*light blue*), 50% change (*yellow*), and 60% change (*black*)

Table 4.1 Number of vertices, edges, and average degree in the PPI networks studied here

Network	# Vertices	# Edges	Average degree
S. cerevisiae	2,345	5,609	4.78
E. coli	1,441	5,871	8.14
H. pylori	687	1,351	3.93

chosen in a way to achieve an average degree of 7 as per the Yeast PPI network. Figure 4.8 compares five independent graphs generated by the Random Geometric Model and the Yeast PPI network.

2.3.1 Generalized Duplication Model: The Importance of Seed Network Selection

The Generalized Duplication Model is perhaps unique among the random network generation models due to its "dependency" on the initial, seed network used. As observed in [16], the seed network shapes the topology of the Generalized Duplication Model and the distributions of all network features, perhaps with the exception of the degree distribution, significantly.

A particularly promising way to pick a seed network is due to the following observation [16]: the Duplication Model is unlikely to generate "large" cliques.[1] On the other hand, the Yeast PPI network includes a clique with 10 nodes, which, as a result, must be included in the seed network. There are other smaller cliques in Yeast PPI network which may need to be represented in the seed graph – this is achieved by adding to the network a single independent clique with 7 nodes. In [16], these two cliques are highly connected in the seed network, which also includes a few additional nodes sparsely connected to the two cliques (the total number of nodes was 50) so that the normalized degree distribution of the Yeast PPI network was similar to that of the seed graph. This "ensures" that the (normalized) degree distribution of the Yeast PPI network as well as its clique frequency distribution (which turns out to be an important determinant of the overall graphlet distribution) are similar to that of the seed graph.

There are two additional parameters associated with the Generalized Duplication Model: p, the edge maintenance probability and r, the edge insertion probability to determine the (asymptotic) degree distribution and the average degree of the generated network. In [16], these parameters are chosen to be $p = 0.365$ and $r = 0.12$ so that the degree distribution of the Duplication Model matches that of the Yeast PPI network. Here, we report on five independently generated networks through the Generalized Duplication Model with these parameters, compared against the Yeast PPI network (see Fig. 4.9). Under all these distributions, the Yeast PPI network seems to be very similar to those produced by the Generalized Duplication Model.

[1] By large cliques we mean its size should be bigger than 5 or 6 nodes.

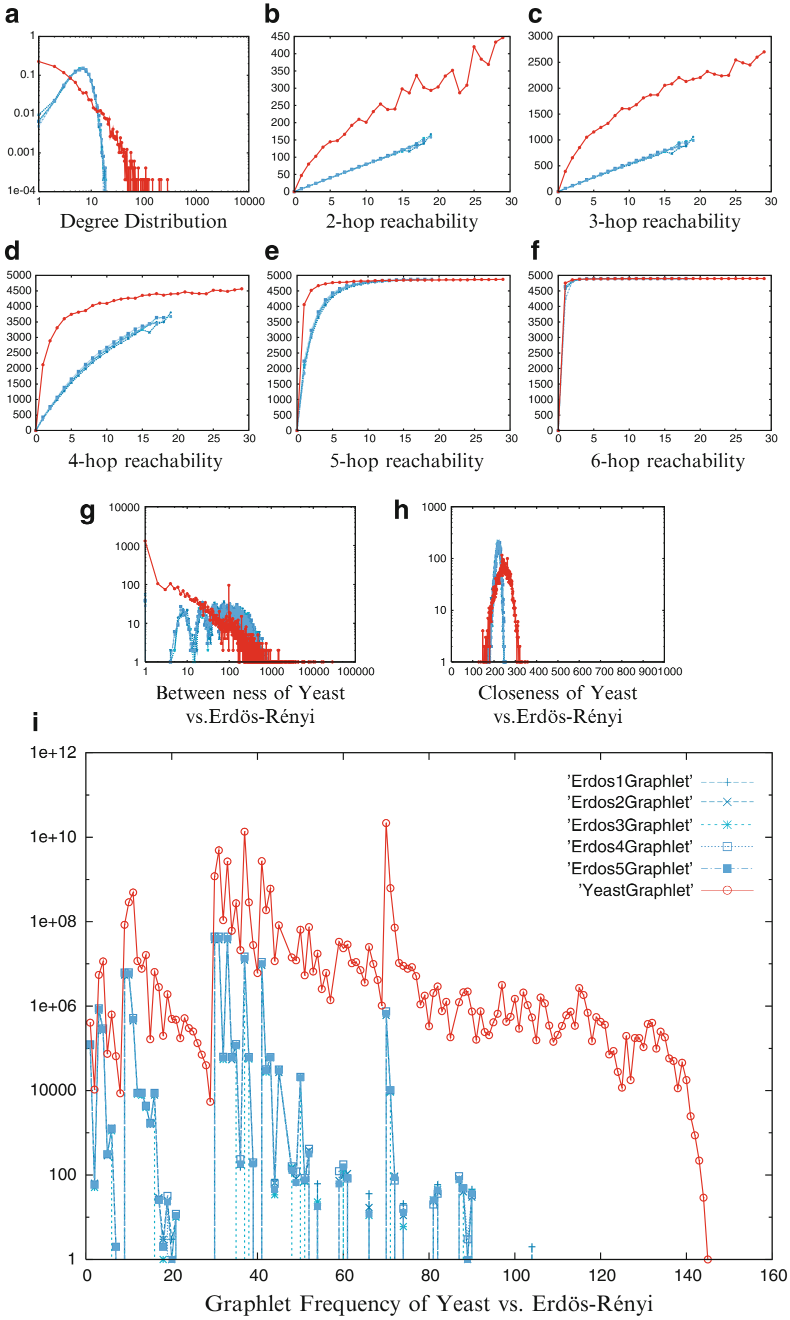


Fig. 4.6 The topological properties of the Erdös–Rényi model (*blue*) compared to that of the CORE yeast network (*red*). The degree distribution, the *k*-hop reachability, graphlet, betweenness, and closeness distributions of both networks are shown. The values are obtained by five independent runs of the duplication model

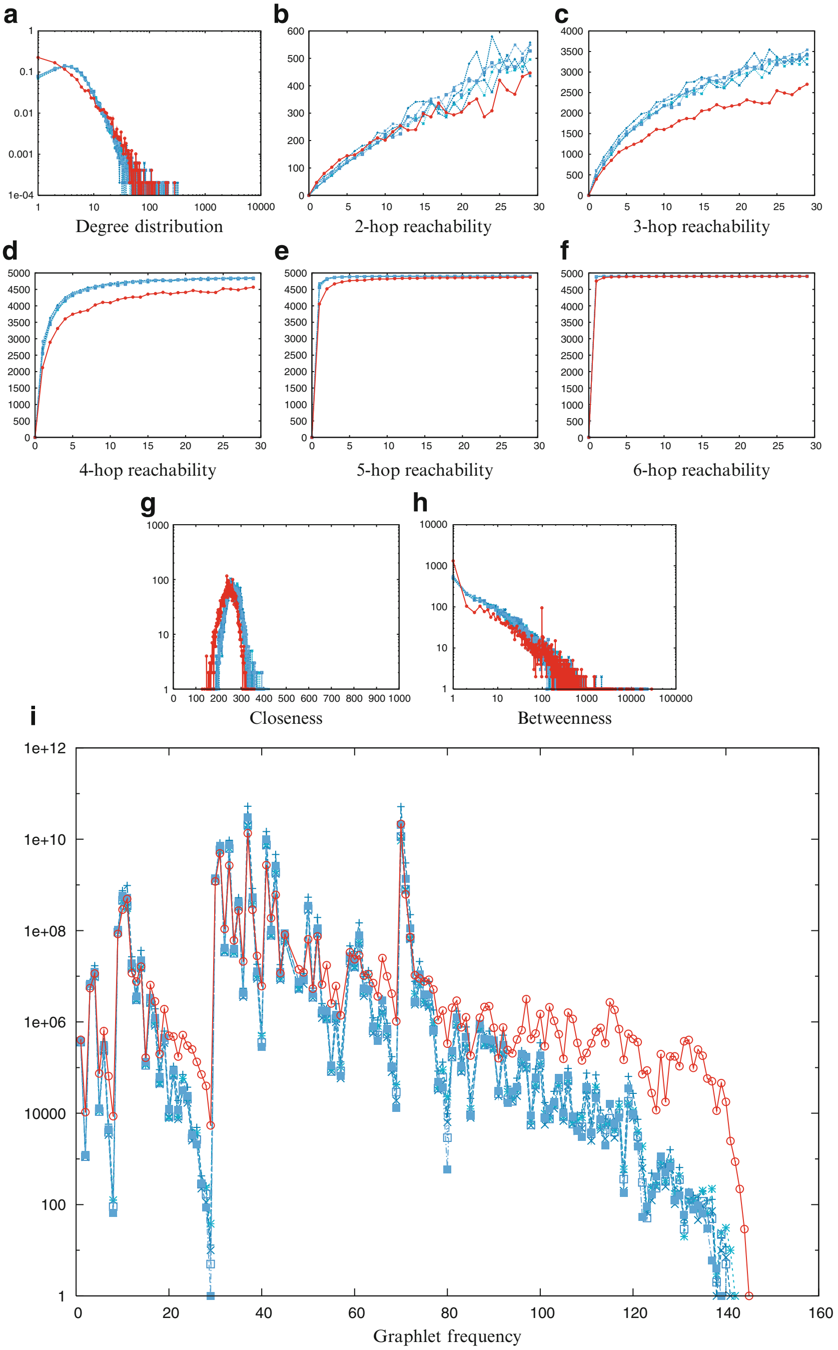


Fig. 4.7 The degree distribution, the *k*-hop reachability, the graphlet, closeness, and betweenness distributions of the yeast PPI (*red*) network against five independent runs of the preferential attachment model (*blue*)

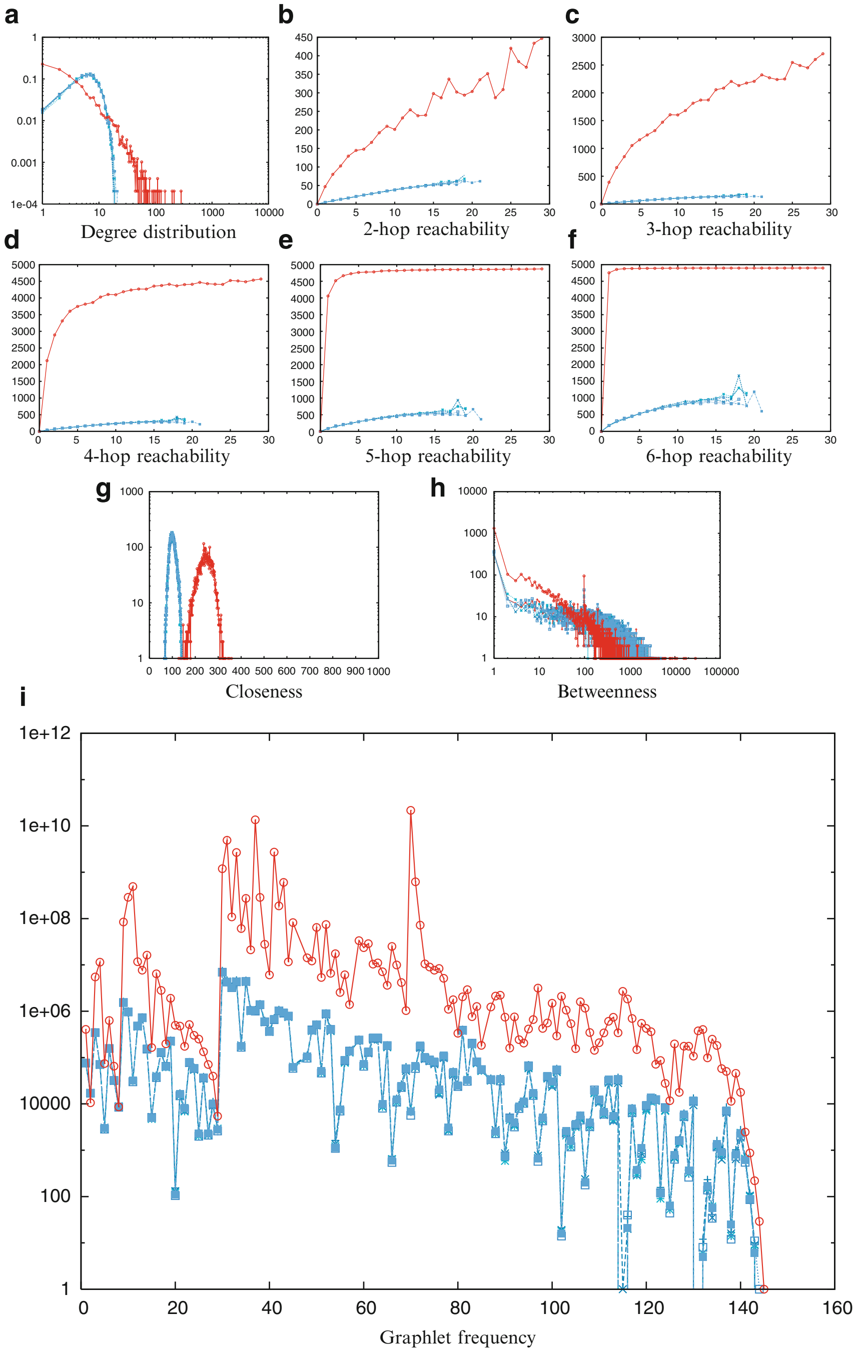


Fig. 4.8 The degree distribution, the *k*-hop reachability, the graphlet, closeness, and betweenness distributions of the yeast PPI (*red*) network against five independent runs of the random geometric model (*blue*)

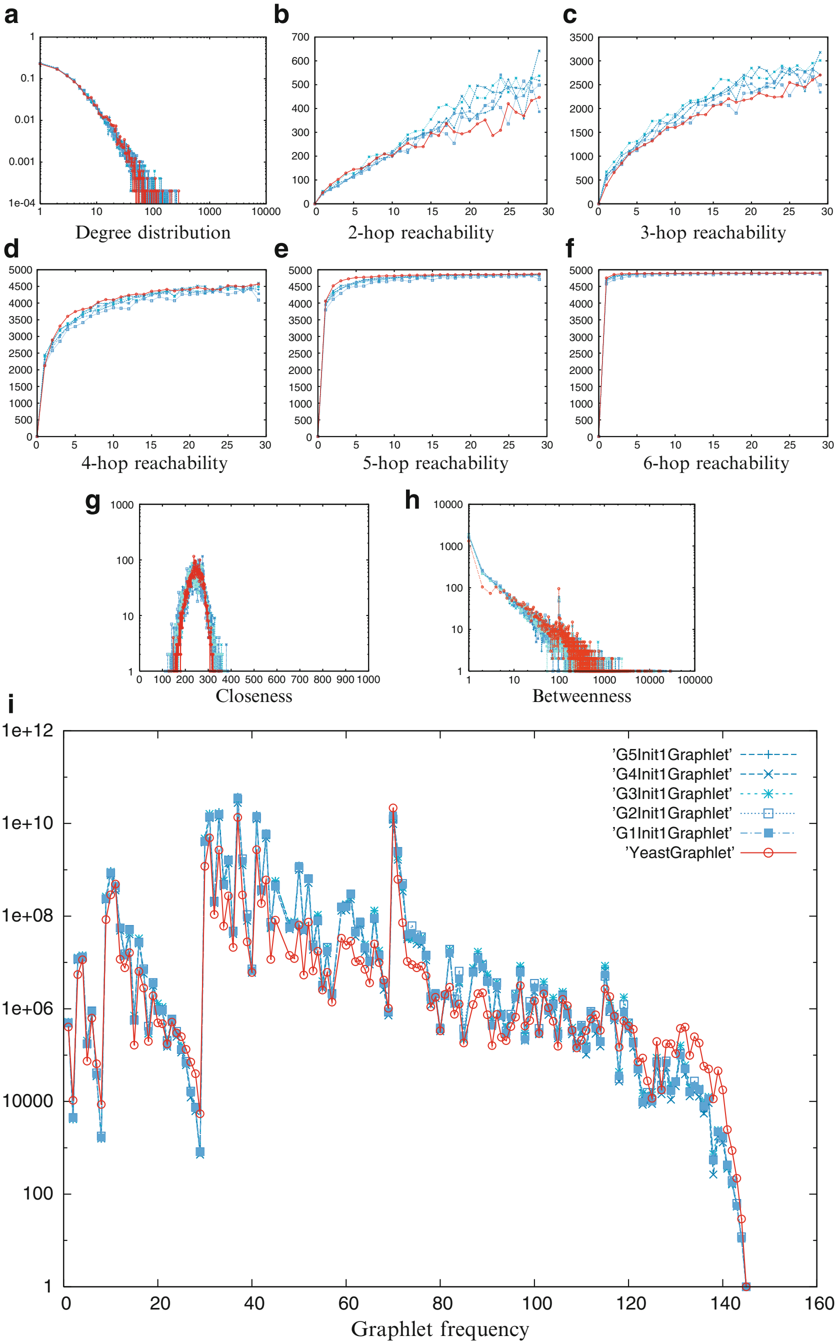


Fig. 4.9 The degree distribution, the *k*-hop reachability, the graphlet, closeness, and betweenness distributions of the yeast PPI (*red*) network against five independent runs of the generalized duplication model (*blue*)

As it can be seen in Figs. 4.7–4.9, it is clear that the *Generalized Duplication Model* with right choice of initial seed network gives results much closer to the real Yeast PPI network.

2.3.2 Comparing PPI Networks of Varying Size

Comparing natural PPI networks or emulated networks of varying sizes is a significant challenge: neither the graph isomorphism-based measures nor many of the feature-based comparisons are useful in this context. It is of crucial importance to "normalize" distributions of network features with the sizes of the networks compared. One such attempt was presented by Hajirasouliha et al. [22] where PPI networks of different organisms and networks generated by proteome emulation models of different number of nodes and edges, were compared with respect to the (normalized) treelet or normalized graphlet distributions as described earlier. Here, the normalization is performed with respect to the total number of treelets or graphlets of a particular size, and not with respect to the size of the network.

In this section, we compare the treelet distributions of the PPI networks of three species: Yeast, *E.coli*, and *H.pylori* against both the Generalized Duplication Model – which all seem to be quite similar to each other despite their highly varying sizes. These networks are also compared against the Preferential Attachment Model, which seems to be fundamentally different from the others (note that Geometric or Erdos–Renyi type network models provide even more variations).

The algorithm for counting the noninduced occurrences of treelets was introduced in [22] which is based on the color-coding technique [2] and was implemented to count all tree topologies of up to ten vertices. The results, originally presented in [22], are presented in Fig. 4.10, where for each treelet (again indexed with respect to robustness of the treelets) the normalized number of occurrences in each network is plotted.

2.4 PPI Network Emulation: Further Challenges

As demonstrated earlier, the Generalized Duplication Model seems to provide the most accurate emulator of available PPI networks among all random graph models. Unfortunately, even the Generalized Duplication Model does not provide an ultimate answer for PPI network emulation. We have demonstrated earlier that certain dense subgraphs (those involving many internal interactions, such as cliques) cannot be captured that well by some of the random graph models introduced. For example, $K_{3,3}$ cannot exist in the networks generated by the 3-D Geometric Random Model. Similarly, the seed networks used for the Generalized Duplication Model need to be dense to emulate a PPI network that contains a large clique. In what follows, we will have a closer look at the presence of dense subgraphs in networks

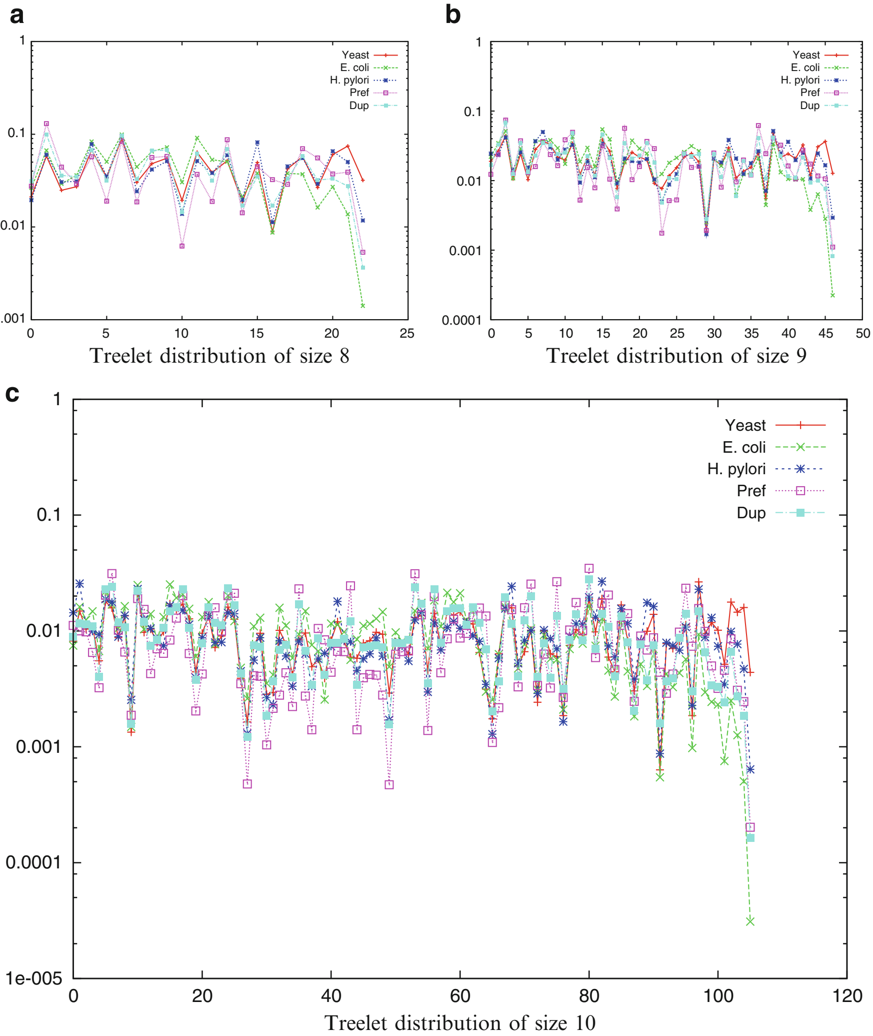


Fig. 4.10 Treelet distribution of the yeast (*red*), *H.pylori* (*blue*), *E.coli* (*green*) PPI networks and the preferential attachment model (*pink*), the generalized duplication model (*cyan*)

generated by random models in comparison to typical dense networks that not only exist in natural PPI networks but also have functional significance.

Let $d(G)$ denote the density of a connected graph G of n vertices which can be defined as the ratio of the number of edges in G and the maximum possible number of edges in a graph with n vertices (i.e., $(n^2 - n)/2$). G is said to be α-dense if $d(G) \geq \alpha$, and dense in general if $\alpha \geq 1/2$. Colak et al. [28] give an efficient method to find all induced dense subgraphs in a (connected) network. This method runs in time polynomial with the number of dense subgraphs in the network (of any size)

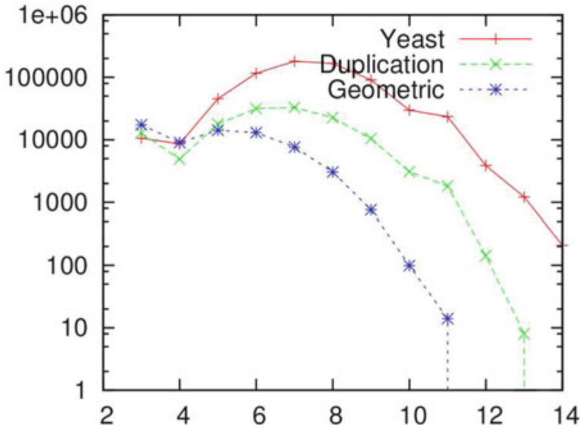


Fig. 4.11 Total number of dense subgraphs (with density ≥ 0.85) with n nodes ($n \in \{3,\ldots,14\}$) as a function of n in the Yeast PPI network as well as networks generated by the geometric random model and the gene duplication-based model. The parameters of the models are set to emulate the Yeast PPI network. The specific colors used are the geometric random model (*blue*), the gene duplication-based model (*green*), and yeast (*red*)

as well as the network size. The algorithm is based on the observation that each dense graph with k vertices includes a dense subgraph with $k-1$ vertices; thus bigger dense subgraphs can be constructed inductively by adding to smaller dense subgraphs, one new node at a time, and observing whether the network stays dense.

Figure 4.11 (originally presented in [28]) shows the total number of dense graphlets (with density $\geq$0.85) with n nodes varying as a function of n (varying between 3 and 14) – for the Yeast PPI network as well as the specific networks generated by the Geometric Random Model and the Generalized Duplication Model – whose parameters were set to best emulate the Yeast PPI network. As shown in Fig. 4.11, there is a large gap between the total number of dense graphlets in the Yeast PPI network and the random networks generated by the Geometric Random Model and the Gene Duplication-Based Model. Although the number of dense graphlets for $n = 6$ is consistent with an earlier study, Fig. 4.11 shows substantial difference for $n > 6$ between the Generalized Duplication Model and the Yeast PPI network, especially for $n \geq 8$, where there is a sevenfold difference. Furthermore, there is a 50-fold (or more) difference between Geometric Random Graph Model and the Yeast PPI network for $n \geq 8$. More drastically, the Geometric Random Model includes no dense graphlets with $n = 12$ nodes and the Gene Duplication-Based Model includes no dense graphlets with $n = 14$ nodes. These figures implies that no random graph generation model is suitable for emulating the growth of PPI networks, at least for the purposes of capturing dense graphlet distributions. It remains an open problem to modify, especially the seed network selection of, the Duplication-Based Models so as to better capture the distribution of denser graphlets.

3 Discussion

Although it is tempting to emulate the topological features of the available PPI networks through random network models, even testing or measuring the similarity of two networks remains to be a challenging problem. The natural measure based on approximate graph isomorphism is not only hard to compute but also is of limited relevance when comparing networks of different sizes. Much of the literature on comparing PPI networks and random network emulators thus relies on comparing (distributions) of topological features such as the degree distribution, k-hop degree distribution, betweenness, and closeness distributions. In addition, comparing the (normalized) number of specific subgraphs, induced or noninduced, is gaining popularity. Although counting the occurrences of specific "graphlets" is a challenging problem, newly emerging algorithms manage PPI networks reasonably well due to their sparse nature. Based on available studies of these network feature distributions, the Generalized Duplication Model seems to provide the best emulator among all – with a caveat: no available random network generation model, including the Generalized Duplication Model seems to capture the "dense" graphlet distributions observed in available PPI networks. Thus, it remains an open problem to devise random network generators that can produce sparse networks which include a (significantly) larger number of dense subgraphs.

References

1. W. Aiello, F. Chung, and L. Lu. A random graph model for power law graphs. In *Proceedings of ACM STOC*, pages 171–180, 2000.
2. Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995.
3. A.-L. Barabási and R. A. Albert. Emergence of scaling in random networks. *Science*, 286: 509–512, 1999.
4. G. Bebek, P. Berenbrink, C. Cooper, T. Friedetzky, J. Nadeau, and S.C. Sahinalp. The degree distribution of the general duplication models. *Theoretical Computer Science*, 369(1–3): 239–249, 2006.
5. G. Bebek, P. Berenbrink, C. Cooper, T. Friedetzky, J. Nadeau, and S.C. Sahinalp. Topological properties of proteome networks. In *Proceedings of RECOMB satellite meeting on System Biology*. LNBI,Springer, 2005.
6. A. Bhan, D. J. Galas, and T. G. Dewey. A duplication growth model of gene expression networks. *Bioinformatis*, 18:1486–1493, 2002.
7. B. Bollobás, O Riordan, J. Spencer, and G. Tusanády. The degree sequence of a scale-free random graph process. *Random Struct. Algorithms*, 18:279–290, 2001.
8. F. Chung, L. Lu, and D.J. Galas. Duplication models for biological networks. *Journal of Computational Biology*, 10:677–687, 2003.
9. C. Cooper and A. Frieze. A general model of webgraphs. *Random Struct. Algorithms*, 22: 311–335, 2003.
10. M. Rasajskim, D. J. Higham, and N. Przulj. Fitting a geometric graph to a protein-protein interaction network. *Bioinformatics*, 8:1093–1099, 2008.
11. E. De Silva and M.P.H. Stumpf. Complex networks and simple models in biology. *Journal of the Royal Society Interface*, 2:419–430, 2005.

12. M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM*, pages 251–262, 1999.
13. R. Ferrer i Cancho, and C. Janssen. The small world of human language. In *Proceedings of Royal Society of London B*, volume 268, pages 2261–2266, 2001.
14. Michael R. Garey and David S.Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
15. J. Han, D. Dupuy, N. Bertin, M. Cusick, and M. Vidal. Effect of sampling on topology predictions of protein-protein interaction networks. *Nature Biotech*, 23:839–844, 2005.
16. F. Hormozdiari, P. Berenbrink, N. Przulj, and S.C. Sahinalp. Not all scale free networks are born equal: the role of the seed graph in ppi network emulation. In *Proceedings of RECOMB satellite meeting on System Biology*, 2006.
17. B. Kahng S. Redner J. Kim, P.L. Krapivsky. Infinite-order percolation and giant fluctuations in a protein interaction network. *Phys. Rev. E 66*, 2002.
18. H. Jeong, S. Mason, A.-L. Barabasi, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411:41, 2001.
19. J. Kleinberg, R. Kumar, PP. Raphavan, S. Rajagopalan, and A. Tomkins. The web as a graph: Measurements, models and methods. In *Proceedings of COCOON*, pages 1–17, 1999.
20. R. Kumar, P. Raghavan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *Proceedings of FOCS*, pages 57–65, 2002.
21. D. G. Corneil, N. Przulj, and I. Jurisica. Modeling interactome: Scale-free or geometric? *Bioinformatics*, 150:216–231, 2005.
22. N. Alon, P. Dao, I. Hajirasouliha, F. Hormozdiari, and S.C. Sahinalp. Biomolecular network motif counting and discovery by color coding. *Bioinformatics*, 24: i32–i40, 2008.
23. D. J. Higham, O. Kuchaiev, M. Rasajski and N. Przulj. Geometric de-noising of protein-protein interaction networks. *Plos Computationtal Biology*, 5, 2009.
24. Ohno. *Evolution by gene duplication*. Springer, 1970.
25. P. Dao, A. Schönhuth, F. Hormozdiari, I. Hajirasouliha, S.C. Sahinalp, and M. Ester. Quantifying systemic evolutionary changes by color coding confidence-sored ppi networks. In *Proceedings of the WABI 2009*, pages 37–48, 2009.
26. R. Pastor-Satorras, E. Smith, and R.V. Sole. Evolving protein interaction networks through gene duplication. *Journal of Theoretical biology*, 222:199–210, 2003.
27. T. Przytycka and Y.K. Yu. Scale-free networks versus evolutionary drift. *Computational Biology and Chemistry*, 28:257–264, 2004.
28. F. Moser, A. Schnhuth, J. Holman, M. Ester, R. Colak, F. Hormozdiar, and S.C. Sahinalp. Dense graphlet statistics of protein interaction and random networks. In *Proceedings of the Pacific Symposium on Biocomputing 2009*, pages 190–202, 2009.
29. A.-L. Barabsi, R.A. Albert. Topology of evolving networks: local events and universality. *Phys. Rev. Lett.*, 85:5234, 2000.
30. S. Redner. How popular is your paper? an empirical study of the citations distribution. *European Physical journal B*, 4:131–134, 1998.
31. Erdös and Rényi. On random graphsI. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.
32. H. A. Simon. On a class of skew distribution functions. *Biometrika*, 42:425440, 1955.
33. A.N. Samukhin, S.N. Dorogovstev, J.F.F. Mendes. Structure of growing networks with preferential linking. *Phys. Rev. Lett.*, 85:4633, 2000.
34. J.F.F. Mendes, S.N. Dorogovstev. Evolution of networks with aging of sites. *Phys. Rev. E*, 62:1842, 2000.
35. R. Tanaka and et al. Some protein interaction data do not exhibit power law statistics. *FEBS Letters*, 579:5140–5144, 2005.
36. A. Vázquez, A. Flammini, A. Maritan, and A. Vespignani. Modelling of protein interaction networks. *Complexus*, 1:38–44, 2003.
37. A. Wagner. The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes. *Molecular Biology and Evolution*, 18:1283–1292, 2001.

38. D.J. Watts. *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton University Press, 1999.
39. D.J. Watts and S.H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393: 440–442, 1998.
40. I. Xenarios and et al. Dip, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Research*, 30:303–305, 2002.
41. G. Yule. A mathematical theory of evolution based on the conclusions of dr. j.c. willis. *Philos. Trans. Roy. Soc. London (Ser. B)*, 213, 1925.

Chapter 5
Biological Network Alignment

Shahin Mohammadi and Ananth Grama

Abstract Recent experimental approaches to high-throughput screening, combined with effective computational techniques have resulted in large, high-quality databases of biochemical interactions. These databases hold the potential for fundamentally enhancing our understanding of cellular processes and for controlling them. Recent work on analyses of these databases has focused on computational approaches for aligning networks, identifying modules, extracting discriminating and descriptive components, and inferring networks. In this chapter, we focus on the problem of aligning a given set of networks with a view to identifying conserved subnetworks, finding orthologies, and elucidating higher level organization and evolution of interactions. Network alignment, in general, poses significant computational challenges, since it is related to the subgraph isomorphism problem (which is known to be computationally expensive). For this reason, effective computational techniques focus on exploiting structure of networks (and their constituent elements), alternate formulations in terms of underlying optimization, and on the use of additional data for simplifying the alignment process. We present a comprehensive survey of these approaches, along with important algorithms for various formulations of the network alignment problem.

1 Introduction

The emergence of high-throughput screening techniques coupled with computational approaches to network reconstruction and inference, have resulted in large databases of biochemical interactions. These interactions can be effectively

A. Grama (✉)
Department of Computer Science, Purdue University, 305 N. University Street,
West Lafayette, IN 47907-2107, USA
e-mail: ayg@purdue.edu

M. Koyutürk et al. (eds.), *Functional Coherence of Molecular Networks in Bioinformatics*, DOI 10.1007/978-1-4614-0320-3_5,
© Springer Science+Business Media, LLC 2012

analyzed to gain novel insights into cellular processes, and identify suitable approaches to controlling these processes. One such analysis technique relies on aligning multiple networks with a view to understanding the conserved functional and organizational principles of biological systems.

The problem of network alignment takes as input one or more networks and establishes correspondences between nodes in the network(s) that optimize a given objective function. Here, the objective function is designed to reflect the conservation of interactions across two or more species. Such analysis for sequences (amino-acid or nucleotide sequences) has been used to great effect in understanding the structure and function of biomolecules. The basic idea that conserved subsequences are likely to share structure, function, and evolutionary trajectories provides the basis for large classes of computational techniques. Network alignment can similarly be used for identifying functionally coherent machinery – "shared function is likely to reflect in aligned subcomponents," and vice-versa. This principle can be used to "project" or "transfer" interaction machinery across organisms that share corresponding function, and to identify latent orthologies among constituent elements. Building further on this premise, alignment can also provide valuable insights into evolutionary trajectories and specialization. As more interaction databases become available, network alignment provides an essential tool for identifying descriptive (and discriminative) components corresponding to the phenotype. Clearly, network alignment in its various forms discussed in this chapter, is an important analysis tool for biochemical pathways.

Given extensive computational infrastructure for sequence alignment, it is natural to examine the relationship between sequence and network alignment [1,2]. In this context, the two key questions relate to models and methods. Models provide a formal framework for alignment problems – namely, they quantify the fitness of an alignment (when one alignment is better than the other) and its significance (how likely is an alignment to correspond to biologically relevant artifacts). Methods, on the other hand, use these models to arrive at desirable alignments and their significance scores. For sequence analysis, BLAST is one of the most commonly used alignment methods, which relies on statistical measures like p-values to quantify significance of alignments. It is easy to see that sequences are special cases of networks – networks with linear connectivity. It follows then that the problem of alignment of *general* networks is at least as hard as sequence alignment (recall that most formulations of multiple sequence alignment are classified into the family of NP-Hard problems – problems for which subexponential time algorithms are not known).

An instance of the network alignment problem for two networks (or a network with itself) is the subgraph isomorphism problem. This problem relates to the identification of the largest common subnetwork of the two given networks. The subgraph isomorphism problem is known to belong to the class of NP-Hard problems as well. The consequent exponential time complexity of solving this problem renders general combinatorial approaches to solving this problem intractable for biochemical networks of interest. Consequently, the problem has motivated a rich class of models and methods that rely on applications' characteristics to solve the problem.

A second important aspect of the problem relates to quantification of significance values associated with alignments. The significance of an alignment quantifies the likelihood of obtaining the quality of an alignment by chance only. The smaller the likelihood, the more significant (hence, more likely to be biologically relevant) the alignment. Traditional approaches to quantifying significance rely either on analytical formulations or on simulations. The state-of-the-art in analytical modeling of networks is in relative infancy. Simulation based methods, on the other hand, suffer from slow convergence and high computational cost. Consequently, quantification of significance poses intriguing challenges, that continue to be investigated.

In the rest of this chapter, we will provide an overview of models, methods, validation techniques, and key data sources for alignment of biochemical networks.

2 Definitions and Notations

Biological networks are often modeled as graphs consisting of vertices (or nodes) and edges (or arcs). Formally, a graph G is defined as $G=(V,E)$, where V is a finite set of vertices and E is a finite set of edges, such that $E \subseteq (V \times V)$. A graph can be either *directed* or *undirected*. In an undirected graph, edges define a symmetric relation among graph vertices, meaning that a relation between vertices v_i and v_j also implies the same relation between v_j and v_i. In directed graphs, relations are not implicitly symmetric. In a directed graph with an edge (v_i, v_j), we refer to v_i and v_j as the source and sink of the edge, respectively.

Graphs can be represented in different ways (i.e., using different data structures). While these representations are logically equivalent, depending on the operations on the graph, some are more computationally efficient than others. One of the commonly used representations is the node adjacency matrix – given a graph G with n nodes, we construct a matrix $\mathbf{A}$ of dimension $n \times n$, in which entry a_{ij} specifies whether there exists an edge between nodes v_i and v_j in G.

In many applications, one also needs to encapsulate additional information about vertices (entities) or edges (relations) of the input network. *Attributed graphs* allow for embedding such information in the graphs. An *edge attributed* (also known as *edge colored*) graph is the one in which the edges have additional information, while the *node attributed* (also known as *node colored*) graph has additional information about the nodes. *Edge weighted graphs* or simply *weighted graphs* are special cases of edge attributed graphs, in which every edge has a real valued attribute (or weight). These weights can be stored in the adjacency matrix of graph G by allowing a_{ij} to store the weight of edge $(v_i, v_j) \in E$. Another way to encapsulate node (or edge) attributes, which is especially useful if we have multiple attributes, is to attach vectors to graph vertices and/or edges.

There are different kinds of biological data that are represented using graphs. *Protein–protein interaction (PPI)* networks are often used in a variety of analyses tasks. In these networks, each node represents a protein and each edge indicates a physical interaction between a pair of proteins. A PPI network can be modeled using an undirected, weighted or unweighted, graph. In the former, the weight usually indicates the probability or confidence of the PPI.

Metabolic networks are often used to understand chemical compositions and reactions. There are two complementary representations of a metabolic network, both of which rely on directed graphs. In the first one, each vertex represents a chemical compound (substrate), and there is an edge between a pair of vertices if they occur (either as substrates or products) in the same chemical reaction. In the second representation, each vertex represents a chemical reaction catalyzed by an enzyme E_i, and there is an edge between any pair of vertices i and j, representing enzymes E_i and E_j, respectively, if they share at least one chemical compound, either as substrate or as product. In other words, if E_i catalyzes a reaction in which compound A is produced, and E_j takes A as a substrate.

Other data, relating to signaling, gene regulations, and lethal interactions are also modeled as graphs. Cell signaling corresponds to the basic communication network of a cell. It governs how a cell perceives and responds to its physio-chemical environment, regulates basic processes such as development, growth and repair (at the tissue level), response to stress, etc. Nodes in these networks correspond to biomolecules (or complexes thereof) and edges correspond to signals. Nodes and edges in these networks are typically labeled to indicate the spatial localization, nature of signals, and type of biomolecules. Gene regulatory networks (GRNs) represent the interactions between genes (through their respective products, which are often not explicitly annotated in the network). Individual nodes correspond to genes and edges correspond to their regulatory roles. An edge from node (gene) i to j implies a regulatory relationship. Since a regulatory link may be positive (up-regulation) or negative (down-regulation), edges are sometimes categorized into up-or down-regulatory edges. GRNs are often modeled as networks of reactions – each modeled using an ordinary differential equation (based on chemical kinetic models). In such networks, rate constants are used to annotate edges. Other models such as Boolean Networks (genes, or nodes are restricted to binary states, that is, they can be on or off, and edges change the state of downstream nodes) and Bayesian Networks (recognizing the stochastic nature of the regulation process).

More recently, data from synthetic genetic arrays have been represented as networks coding synthetic lethality. Synthetic lethality refers to the observation that a combination of two or more gene mutations leads to cell death, while a single mutation to either of these genes does not. In synthetic lethality networks, nodes correspond to genes and edges reflect the existence of a synthetic lethal interaction between the two genes.

2.1 Network Alignment Problems

Given a set of graphs $\mathcal{G} = \{G_1, G_2, \ldots, G_k\}$, an alignment corresponds to a proper mapping between the nodes of input networks that maximizes the similarity between mapped entities. The *pairwise network alignment problem* is a special case of this problem with two input networks. Network alignment, in its general form, is a computationally hard problem, since it can be related to the subgraph-isomorphism problem, which is known to be NP-complete. Effective techniques for solving this problem rely on suitable formulations of the alignment problem, use of heuristics to solve these problems, or on the use of alternate data to guide the alignment process.

At a high level, the network alignment problem can be classified as *local alignment* or *global alignment*. The former is a relationship over a subset of the nodes in $V = \{V_1 \cup V_2 \ldots \cup V_k\}$, while the latter is defined as partitioning all nodes in V into disjoint subsets, also known as equivalence classes [2]. Global network alignment can be further classified into *one-to-one*, in which every subset has exactly n nodes, one from each input network, and *many-to-many*, in which the subsets are not restricted to have exactly one node from each input network.

The biological interpretation of the local alignment problem is that each subset of aligned nodes represents a conserved module. In a global one-to-one alignment, nodes in each subset can be interpreted as functional orthologs, while in many-to-many network alignment, each subset is a classification of all possible functional orthologs in given species into an equivalence class.

We start by denoting the set of all possible alignments as $\mathscr{A}$. It is common to represent each network alignment $A \in \mathscr{A}$ using an *alignment graph*, $G_A = (V_A, E_A)$, where every node in the alignment graph represents an equivalence class, while each edge represents a relationship between a pair of equivalence classes. To define the network alignment problem formally, we also need to define an *alignment scoring function*, $\phi : \mathscr{A} \rightarrow \Re$, which assigns to each alignment $A \in \mathscr{A}$, a real fitness value. Given an alignment scoring function, the global network alignment problem is formally defined as finding the maximum score global network alignment A_{opt}, while the local network alignment problem is defined as finding a set of maximal score local network alignments. The core of any alignment algorithm consists of an *alignment scoring function* together with a *search, or optimization method.*

Before we discuss alignment algorithms, we also introduce a general form for the *node scoring function*, $S : \{V_1 \cup V_2 \ldots \cup V_k\} * \{V_1 \cup V_2 \ldots \cup V_k\} \rightarrow \Re$, which assigns a similarity score to each pair of nodes in the input networks. Different node similarity functions have been proposed, based on the node attributes, as well as the local network topology around each node. An example of the former case is the BLAST score of the protein sequences corresponding to a pair of given nodes, while an example of the latter case is the scoring function proposed by Kuchaiev et al. [3], in which they use a vector representing the number of *graphlets* that each node takes part, to compare the topological similarity around each node.

3 Algorithms and Methods

Alignment problems have been modeled as diverse optimization problems, based on the underlying applications. In this section, we describe the mathematical models underlying these variants of alignment problems and discuss algorithms for these problems. An important and difficult problem associated with these algorithms is their validation. This difficulty stems from the noisy, incomplete, and statistically skewed nature of underlying data. We conclude the discussion in this section with an overview of validation techniques and databases available for analyses and validation.

3.1 Local Alignment

Local alignment corresponds to a relationship defined over a subset of vertices in the input networks. It is often used to extract conserved substructures (modules, pathways, complexes) from a set of species. A number of algorithms have been proposed for local alignment. We provide an overview of these methods in this section.

3.1.1 The Blast Family: PathBlast, NetworkBlast, and NetworkBlast-M

PathBlast [4], proposed by Kelley et al. [5], was among the first attempts at network alignment, with the goal of identifying conserved pathways in a pair of species. The method identifies high-scoring alignments between pairs of pathways, one from each input network, such that proteins in the first pathway map to their putative homologs in the same order in the second pathway. To accomplish this, PathBlast initially builds an alignment graph (see Sect. 2), where edges can be either a *match*, *gap*, or a *mismatch* edge. Let v_i^1 and v_i^2 denote the nodes from first and second species, respectively, in the equivalence class represented by node v_i in the alignment graph. A *match* edge occurs between nodes v_i and v_j in the alignment graph when v_i^1 and v_j^1 are connected in the first species, and v_i^2 and v_j^2 are connected in the second species. Otherwise, it can be either a *mismatch*, or a *gap* edge. The former occurs when neither v_i^1 and v_j^1, nor v_i^2 and v_j^2 are connected in their corresponding species, and the latter occurs when only one of the protein pairs in one of the species are connected.

The core of the PathBlast algorithm is a log probability score for evaluating each pathway P in the alignment graph. This score is computed by decomposing the pathway similarity score into a vertex scoring fraction and an edge scoring fraction. More formally, the scoring function is defined as follows:

$$S(P) = \sum_{v \in P} \frac{p(v)}{p_{\text{random}}} + \sum_{e \in P} \frac{q(e)}{q_{\text{random}}}. \tag{5.1}$$

Here $p(v)$ represents the probability of true homology between the protein pair from input networks represented by node v in the alignment graph. The quantity $q(e)$ represents the probability that interactions represented by e are real interaction, not false positive interactions. Probabilities p_{random} and q_{random} are evaluated as the expected values of $p(v)$ and $q(e)$, respectively. Using this scoring function, one can find the optimal alignment as the one in which the pathway scoring function is optimized over all pathways up to length L for networks of size n using randomized dynamic programming. The method has an expected time complexity $O(nL!)$, if the input networks are acyclic (i.e., do not contain any cycle).

PathBLAST is available through a web-interface at http://www.pathblast.org/. A user may specify a short protein interaction network for query against a target PPI network from a network database. Protein interactomes of yeast (*Saccharomyces cerevisiae*), the bacterial pathogen (*Helicobacter pylori*), bacterium (*Escherichia coli*), nematode worm (*Caenorhabditis elegans*), fruit fly (*Drosophila melanogaster*), mouse (*Mus musculus*), and human (*Homo sapiens*) are available as target species. The program returns a ranked list of matching paths from the target network along with a graphical view of these paths and the associated overlap.

Sharan et al. [6] extend the idea of PathBlast for extracting conserved protein complexes from a pair of input networks. Their algorithm, NetworkBlast, allows extraction of *all* conserved complexes across networks, as opposed to the single query model of PathBlast. The resulting computational problem is more general and difficult. NetworkBlast has also been generalized to NetworkBlast-M [7] for identifying conserved networks among multiple networks.

Sharan et al. initially evaluate the reliability of PPI and build a weighted network by assigning a confidence value to each interaction. They propose a *logistic regression model*, based on the method proposed by Bader et al. [8], and use the following three random variables to define their logistic distribution:

X_1: Number of times an interaction between the proteins is experimentally observed
X_2: Pearson correlation coefficient of expression measurements for the corresponding genes
X_3: Proteins' small world clustering coefficient.

Using these random variables, the probability of a true interaction T_{uv} is defined as:

$$Pr(T_{uv}|X) = \frac{1}{1+\exp(-\beta_0 - \sum_{i=1}^{3} \beta_i X_i)}, \tag{5.2}$$

where $\beta_0, \ldots, \beta_3$ are parameters of the distribution [6]. They then build an alignment graph, in which each node corresponds to a group of k similar proteins, that is, proteins from different species with BLAST E-values smaller than 10^{-7}. Each edge in the alignment graph represents a conserved interaction between the proteins that occur in its end nodes. An edge is considered conserved if and only if one of the following conditions is met:

- A pair of proteins directly interacts, and all other pairs include proteins with distance at most two in their corresponding networks.
- All protein pairs have distance exactly two in their corresponding networks.
- At least max $\{2, k-1\}$ protein pairs directly interact.

Finally, they devise a scoring scheme based on a likelihood model to fit the subnetwork to the given structure. Given a subset U of the vertices, O_U denotes the collection of all observations on vertex pairs in U, and O_{uv} denotes the set of available observations on the proteins u and v, that is, the set of experiments in which an interaction between u and v was, or was not, observed. Also, let T_{uv} denote the event that two proteins u and v interact, and F_{uv} denote the event that they do not interact.

One may formalize the log-likelihood ratio of a subgraph under a conserved subnetwork model, M_s, and under a null model, M_n, in a single species, as follows:

$$L(U) = \log \frac{Pr(O_U|M_s)}{Pr(O_U|M_n)} = \sum_{(u,v)\in U*U} \log \frac{\beta Pr(O_{uv}|T_{uv}) + (1-\beta)Pr(O_{uv}|F_{uv})}{p_{uv}Pr(O_{uv}|T_{uv}) + (1-p_{uv})Pr(O_{uv}|F_{uv})}, \tag{5.3}$$

where β is a high probability of interaction under the clique model, while p_{uv} is the probability of interaction between proteins u and v under the null model (random graph with the same degree distribution). To find the log-likelihood ratio of multiple complexes across different species, one may sum the log-likelihoods for single species.

Using this scoring function, the problem of identifying conserved subnetworks reduces to one of finding high scoring subgraphs. This problem is known to be NP-hard. Consequently, they adopt a greedy approach to this problem, which is based on an extension of high scoring seeds, similar to the BLAST algorithm. NetworkBlast is available via a web interface at http://www.cs.tau.ac.il/~bnet/networkblast.htm. It can also be downloaded as a stand-alone program from the same website.

3.1.2 MaWISh: Alignment Based on Network Evolution Models

Koyutürk et al. [9,10] propose an evolution-based scoring function, which quantifies the evolutionary distance of any pair of induced subgraphs in the input networks. They use this scoring function to align the input networks (see Box 5.1 for a detailed explanation of their scoring scheme). They reduce the local alignment problem into a *maximum weight induced subgraph problem* (MaWISh). Noting the NP-completeness of this problem by reduction from max-clique, they propose a greedy approach to approximate the solution. They initially match the *hub nodes* and iteratively expand the subgraph in the sparse product graph by adding nodes that share a matching edge with these nodes, to maximize their scoring function.

Koyutürk et al. [14] extend their method to multiple networks, by contracting the global alignment graph and then applying algorithms from frequent itemset extraction. The MaWISh software is currently available for download from http://compbio.case.edu/koyuturk/software/mawish.tar.gz.

Box 5.1: MAWISH network evolution based scoring scheme

A common model of evolution that explains preferential attachment is the duplication/divergence model, which is based on gene duplications [11–13]. According to this model, when a gene is duplicated in the genome, the node corresponding to the product of this gene is also duplicated together with its interactions. A protein loses many aspects of its functions rapidly after being duplicated. This translates to divergence of duplicated (paralogous) proteins in the interactome through elimination and emergence of interactions. Elimination of an interaction in a PPI network implies the loss of an interaction between two proteins due to structural and/or functional changes. Similarly, emergence of an interaction in a PPI network implies the introduction of a new interaction between two noninteracting proteins, caused by mutations that change protein surfaces. Examples of duplication, elimination, and emergence of interactions are illustrated in Fig. 5.1.

Using the duplication/divergence model, Koyutürk et al. [9, 10] propose a novel evolution-based scoring function. Given PPI networks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, a *protein subset pair* $P = \{S_1, S_2\}$ is defined as a pair of protein subsets $S_1 \subseteq V_1$ and $S_2 \subseteq V_2$. Given a pair of graphs G_1 and G_2, any protein subset pair P induces a local alignment $A(G_1, G_2, P) = \{\mathscr{M}, \mathscr{N}, \mathscr{D}\}$ of G_1 and G_2 with respect to similarity score function S (see Sect. 2), characterized by a set of duplications $\mathscr{D}$, a set of matches $\mathscr{M}$, and a set of mismatches $\mathscr{N}$. The biological analog of a *duplication* is the duplication of a gene in the course of evolution. Each duplication is associated with a score that reflects the divergence of function between the two proteins, estimated using their similarity. A *match* corresponds to a conserved interaction between two orthologous protein pairs, which is rewarded by a match score that reflects our confidence in both protein pairs being orthologous. A *mismatch*, on the other hand, is the lack of an interaction in the PPI network of one organism between a pair of proteins whose orthologs interact in the other organism. A mismatch may correspond to the emergence of a new interaction or the elimination of a previously existing interaction in one of the species after the

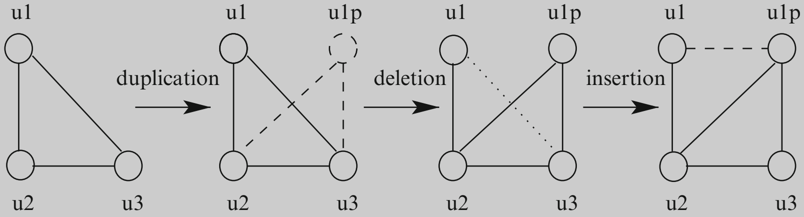


Fig. 5.1. Evolutionary events, and their effects on network topology

(continued)

Box 5.1 (continued)

split, or an experimental error. Thus, mismatches are penalized to account for the divergence from the common ancestor. The formal definitions of these three concepts are as follows:

Definition 5.1. Match, Mismatch, and Duplication Given protein interaction networks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, and a pairwise similarity function S, any protein subset pair $P = (S_1, S_2)$, induces a local alignment $A(G_1, G_2, P) = \{\mathscr{M}, \mathscr{N}, \mathscr{D}\}$, where:

$$\begin{aligned}\mathscr{M} &= \big\{u, v \in V_1, u', v' \in V_2 : 0 < S\left(u, u'\right), \, 0 < S\left(v, v'\right), \\ &\qquad (u, v) \in E_1 \wedge \left(u', v'\right) \in E_2\big)\big\} \qquad (5.4)\\ \mathscr{N} &= \big\{u, v \in V_1, u', v' \in V_2 : 0 < S\left(u, u'\right), \, 0 < S\left(v, v'\right), \\ &\qquad \big((u, v) \in E_1 \wedge \left(u', v'\right) \notin E_2\big) \vee \left((u, v) \notin E_1 \wedge (u', v') \in E_2\right)\big\} \qquad (5.5)\\ \mathscr{D} &= \{u, v \in V_1 : 0 < S(u, v)\} \cup \big\{u', v' \in V_2 : 0 < S\left(u', v'\right)\big\} \qquad (5.6)\end{aligned}$$

Matches $M \in \mathscr{M}$, mismatches $N \in \mathscr{N}$, and duplications $D \in \mathscr{D}$ are associated with scores $\mu(M)$, $\nu(N)$, and $\delta(D)$, respectively. Using this formulation of match, mismatch, and duplication, the evolutionary plausible scoring function to evaluate each network alignment can be defined as follows:

Definition 5.2. Alignment Score Given PPI networks G_1 and G_2, the score of alignment $A(G_1, G_2, P) = \{\mathscr{M}, \mathscr{N}, \mathscr{D}\}$ is defined as:

$$\sigma(A) = \sum_{M \in \mathscr{M}} \mu(M) - \sum_{N \in \mathscr{N}} \nu(N) - \sum_{D \in \mathscr{D}} \delta(D). \qquad (5.7)$$

Equation (5.7) can be used to evaluate the evolutionary distance of any given subset pair in the input networks.

3.1.3 Graemlin: Alignment with Equivalence Classes

Flannick et al. [2] propose an alternate method, Graemlin, which improves over previous methods by using heuristics from sequence alignment. They propose a formulation of network alignment, based on *equivalence classes*. In this model, the network alignment problem is posed as follows: given a set of input networks, a network alignment is defined as a set of subgraphs together with a symmetric mapping between the corresponding (aligned) vertices. For the alignment to be unique, this mapping should be transitive, meaning that $A \leftrightarrow B, B \leftrightarrow C \Rightarrow A \leftrightarrow C$; mathematically, such a symmetric-transitive relation is also known as equivalence relation. This definition classifies the aligned vertices into disjoint groups (*equivalence classes*). Each equivalence class consists of proteins evolved from a common ancestral protein, and unlike previous definitions, can contain multiple proteins in same species, also known as paralogs. This formulation allows them to modify the *progressive alignment* method adapted from sequence alignment, and to be able to scale linearly in the number of the networks compared. They also use a heuristic similar to *seed extension* in sequence alignment, to align the input networks efficiently, and to be able to trade-off speed versus sensitivity.

Using this formulation, Flannick et al. propose a scoring function composed of two parts, one to evaluate each equivalence class, and the other to evaluate each edge in the alignment. The former is more straightforward, while the latter is more involved, but provides the opportunity to search for arbitrary module structures. The scoring scheme is similar in both: find the probability distribution defined for two different models, namely the constrained alignment model $\mathscr{M}$ based on a given module structure and random model $\mathscr{R}$, and define the score function as the log-ratio of two probabilities. Equation (5.8) presents the Graemlin scoring function.

$$S = S_c + S_e, \text{ where } \begin{cases} S_c = \log\left(\frac{P_{\mathscr{M}}(c)}{P_{\mathscr{R}}(c)}\right) \\ S_e = \log\left(\frac{P_{\mathscr{M}}(e)}{P_{\mathscr{R}}(e)}\right) \end{cases} \tag{5.8}$$

Scoring of equivalence classes is based on construction of the most parsimonious ancestral history of the proteins in each equivalence class. This construction is based on sequence mutations, insertions, deletions, duplication, and divergence among proteins in each class. The probability of sequence mutations is estimated in a principled manner in their study; other events are determined heuristically. The alignment model M is trained by sampling pairs of proteins from within the same COG [15] group, while the random model R corresponds to picking random pairs in the network (see Flannick et al. [2] supplementary material for a detailed description).

Scoring of alignment edges is based on the concept of an *Edge Scoring Matrix (ESM)*, a symmetric matrix defined over a set of alphabets, Σ, in which every entry in the matrix is a probability distribution over edge weights. Graemlin first assigns alphabets to each equivalence class, then it scores each alignment edge using the cell in the ESM index by the labels assigned to two endpoints of the edge. This approach extends the previous methods in that it is capable of searching for conserved substructures with user-defined structure, not just pathways or complexes.

The next two steps use the score function to align a pair of networks, and to extend this approach to multiple alignment. Graemlin mimics the *seed extension* method, meaning that it tries to find a proper set of candidate seed vertices, and then extends them greedily. Unlike MaWISh, the seed vertices are chosen in a way that does not impose special topology (clique-like) on the subgraph structure. Seed selection in Graemlin is based on the concept of *d-clusters*, it first selects d-clusters for each node by finding $d-1$ nearest neighbors, where the distance between vertices is defined as the negative log of edge weights. It then finds the pairwise node similarity score of sample mappings between two d-clusters, one from each species, and reports the highest score among them. The d-clusters with mapping scores higher than the user defined threshold T are used as *seeds*. Parameters d and T are adjustable parameters that can be used to trade-off speed versus sensitivity in the algorithm. After computing the seeds, Graemlin greedily expands each equivalence class by coalescing vertices in the frontier of each equivalence class.

An extension of this approach to multiple alignment using an analog of the *progressive alignment* technique, commonly used in sequence alignment. Having an extended phylogenetic tree with species on the leaves, the technique successively aligns the closest pair of networks, and places three new networks in the parent node: one for the alignment network, and two other networks for unaligned subsets of the pair of networks.

Flannick et al. [2] construct ten weighted microbial PPI networks based on the SRINI algorithm [16]. These are publicly available at http://graemlin.stanford.edu/nets.tar.gz. Graemlin1.0 can be freely downloaded from http://graemlin.stanford.edu/graemlin-1.0.tar.gz as a stand-alone application.

3.1.4 Information Theoretic Network Alignment

Yet another method, motivated by information theory, is recently proposed by Chor et al. [17]. The fundamental idea in this method is to devise a computationally tractable measure that computes the disparity between two uniquely labelled graphs G_1 and G_2. This problem is then reduced to finding how many additional bits do we need to encode a graph G_2 given graph G_1 (known as description length of G_2 given G_1). To tackle this problem, Chor et al. impose the following key assumptions:

- *Shortest path conservation*: If a pair of nodes u and v are common in the vertex set of both networks, the length of the shortest path between them in the underlying graph of G_1 and G_2 must be similar.
- *Neighborhood conservation*: If a pair of nodes u in G_1 and v in G_2 are similar in some sense, like homolog proteins in PPI networks, but not identical, then the level one neighborhood of u and v must be highly similar.

Using these assumptions, they developed a measure, $D(G_2|G_1)$, which illustrates the number of additional bits needed for encoding the adjacency list of graph G_2 given graph G_1. This measure is not a distance metric, since it is clearly not symmetric. To devise a metric, they proposed the notion of *relative description length* as follows:

$$\mathrm{RDL}(G_1, G_2) = \frac{\mathrm{DL}(G_1|G_2)}{\mathrm{DL}(G_1)} + \frac{\mathrm{DL}(G_2|G_1)}{\mathrm{DL}(G_2)} \tag{5.9}$$

Armed with the RDL metric, which computes the distance between graphs using an information theoretic method, they tackled the problem of finding conserved regions in networks. Conserved regions are defined as specific vertex-induced subgraphs in each network. More precisely, they first extracted pairs of "similar" nodes in networks, and then used this vertex set to induce subgraphs in corresponding input networks. To find the set of conserved nodes, they started with the set of common vertices, $V' = V_1 \cap V_2$, and proceed by comparing the level d neighborhood of each node $v \in V'$ in networks G_1 and G_2 using RDL metric. Any node that the RDL distance of its level d neighborhoods in G_1 and G_2 exceeds a threshold c will be filtered out from V'. Using V', edge sets E'_1 and E'_2 can be easily found by imposing V' on G_1 and G_2, respectively, and finding the induced subgraph in each network.

Chor et al. [17] successfully apply their method to both metabolic pathways extracted from KEGG database, and on a pair of PPI networks. Since PPI networks do not have unique labeling among networks, they use a heuristic to label the nodes. They define identical nodes in input networks as pairs of nodes in which the BLAST scores of their corresponding proteins have E-value $< \mathrm{e}^{-10}$. This is similar in nature to pruning the state space of mappings from beginning of the algorithm to a very small subset of total possible mappings, namely the most promising ones.

3.1.5 Network Queries: A Supervised Approach to the Network Alignment Problem

Network alignment and integration are focused on de novo discovery of biologically significant regions embedded in a network, based on the assumption that regions supported by multiple networks are functional. In contrast, a supervised approach to conserved module detection relies on a query subnetwork that is previously known to be functional. The objective of such methods is to identify subnetworks in a given network that are similar to the query. Among these methods, MetaPathwayHunter aims to identify metabolic pathways that match a query pathway in a database of pathways [18]. Similarly, Narayanan and Karp [19] aim to find matching pathways in PPI networks based on a match-and-split strategy. Bruckner et al. [20] propose a novel method, named Torque (TOpology-free netwoRk QUErying), which unlike most of the previous methods, does not restrict the topology of query network. Finally, Banks et al. [21] propose an extension of regular expressions on strings to networks, named *network scheme*.

3.2 Global Alignment

Global alignment algorithms aim to find a consistent relationship defined over *all* vertices of the input networks. Global alignment is commonly used to establish functional orthologs across species. A number of models and methods have been proposed for global alignment of networks.

3.2.1 Markov Random Field

One of the early efforts at global alignment of protein interaction networks is due to Bandyopadhyay et al. [22]. This study aims at solving the ambiguity in Inparanoid clusters with more than two proteins, to increase the accuracy of functional ortholog prediction. It is based on the idea that early paralogous proteins (out-paralogs) are more likely to change their interaction patterns and adopt new functions in the cell (for more information, please see Sect. 4.1).

This method uses topological information in PPI networks to maximize the number of conserved interactions to resolve ambiguity. The method relies on a probabilistic model and assigns a binary random variable, z_i to each node i in the alignment graph (representing a pair of aligned nodes in the input graphs). The variable indicates whether the corresponding protein pair represents true functional orthologs or not. Two nodes in the alignment graph, z_i and z_j, are connected if at least one of the protein pairs in the input graph (the protein pair represented by either i or j) are connected, and the other one has a common neighbor (or is also connected). The conditional probability distribution of Z_i can be defined as:

$$P(Z_i|Z_{N(i)}) = \frac{1}{1+\exp\{-\alpha_i + \sum_{j\in N(i)} \beta_{ij} Z_j\}}, \tag{5.10}$$

where $N(i)$ represents the neighbors of node i in the alignment graph. Simply stated, this formulation implies that a pair of proteins represented by node i in the alignment graph are more probable to be true functional orthologs when most of their neighbors are functional orthologs as well. To verify this formulation, one may observe that if we have only two proteins in the cluster, z_i will be 1, and for any pair of proteins in different clusters it is equal to 0. Bandyopadhyay et al. use a training data set to estimate parameters α and β, and use Gibbs sampling to evaluate the distribution function Z. Markov Random Field (MRF) based methods are successfully applied to alignment of protein interaction networks of yeast (*S. cerevisiae*) and fruit fly (*D. melanogaster*) (http://www.cellcircuits.org/Bandyopadhyay2006/).

3.2.2 IsoRank Family: Pairwise IsoRank, IsoRank-M, and IsoRank-N

The basic idea of the IsoRank family of methods, as explained in detail in Box 5.2, is to characterize the similarity of two nodes, v_i in G_1 and v_j in G_2, as a combination of node similarity and topological similarity. This quantity, denoted r_{ij}, is computed for all node pairs. The resulting similarity matrix, R, is used to align the input networks.

Singh et al. [23] propose a pairwise alignment technique based on similarity matrix R. They use a well-known algorithm for graph matching to align a pair of input graphs: they initially built a full weighted bipartite graph (nodes from G_1 in one part, nodes from G_2 in the other part, and edges representing similarity of nodes in G_1 to nodes in G_2). They then compute a *maximum weight bipartite match* using Hungarian algorithm [24], to find the one-to-one global alignment. Since the multiple graph matching problem, unlike bipartite graph matching, is known to be NP-complete, Singh et al. [25] extend this result to multiple network alignment by proposing heuristics for many-to-many alignment of input graphs based on the following greedy approach:

Initialization: Select the edge (v_i^{k1}, v_j^{k2}) with the highest score, where v_i^{k1} and v_j^{k2} are vertices in G_{k1} and G_{k2}, respectively. Initialize a new equivalence class with v_i^{k1} and v_j^{k2} as its initial members.

Expand to other species: In every other species, $\{G_1, \ldots, G_k)\} \backslash \{G_{k1}, G_{k2}\}$, if a node l exists in species G_{kx} such that:

- $R_{il}^{\langle k1,kx \rangle}$ and $R_{jl}^{\langle k2,kx \rangle}$ are the highest scores between l and any node in G_{k1} and G_{k2}, respectively, and
- Both $\beta_1 R_{ij}^{\langle k1,k2 \rangle} \leq R_{il}^{\langle k1,kx \rangle}$, and $\beta_1 R_{ij}^{\langle k1,k2 \rangle} \leq R_{jl}^{\langle k2,kx \rangle}$

then, add it to the primary class. This step ensures that the equivalence class has at most one node from each species.

Heuristic expansion: Add up to $r-1$ nodes from different parts of the graph to the equivalence class. Suppose v (from G_{ky}) is already in the equivalence class. Then, node v' (again from species G_{ky}) is added to the class if $\beta_2 R_{vw}^{\langle ky,kz \rangle} \leq R_{v'w}^{\langle ky,kz \rangle}$, for every node $w \in G_{kz}$ which is already in the equivalence class ($w \neq v$).

Update Remaining: Remove from the alignment graph all of the nodes in the constructed equivalence class, and their corresponding edges.

Here, parameters β_1, $\beta_2 \in (0,1)$ and r are user-defined parameters. Also, $R_{ij}^{\langle p,q \rangle}$ represents the similarity between node v_i from species p, and node v_j from species q.

Liao et al. [26] propose an alternate heuristic for multiple alignment of networks. Their method, called IsoRankN (IsoRank-Nibble), is similar in concept to *PageRank-Nibble*, which approximates the Personalized PageRank vector. This approach constructs a full weighted k-partite graph with pairwise similarity scores as the weight on edges, and use a method based on spectral clustering to cluster the graph into low-conductance sets (similar to partitioning the graph into maximal weight subgraphs). All versions of IsoRank are available for download from http://groups.csail.mit.edu/cb/mna/.

3.2.3 Graemlin Family: Graemlin2.0

Flannick et al. [27] extend the concepts underlying Graemlin 1.0 by incorporating a general scoring framework. This framework is based on a user defined *feature vector*, and a *weight function* that can be learned from a set of true alignments. They also propose a hill-climbing method in Graemlin 2.0 and use this scoring function to align the input networks globally.

Box 5.2: IsoRank Algorithm

The core of all *IsoRank*-based algorithms is a method for computing the similarity matrix R, representing the functional similarity scores between any pair of nodes in two input networks. To compute these similarity scores, Singh et al. [23] propose an approach similar to PageRank. The method is based on the notion that a pair of nodes (i, j) represent a good match if the sequences corresponding to nodes i and j align well, and that their respective neighbors are also good matches. This recursive definition leads to the following formal definition of R:

$$R_{ij} = \sum_{v_u \in N(v_i), v_w \in N(v_j)} \frac{1}{|N(v_u)||N(v_w)|} R_{uw}, \tag{5.11}$$

where $N(v_i)$ represents the neighbors of node v_i in the input network. Using a matrix notation, this equation can be rewritten as:

$$R = AR$$

$$A[i,j][u,v] = \begin{cases} \frac{1}{|N(u)||N(v)|}, & \text{if } (i,u) \in E_1 \text{ and } (j,v) \in E_2; \\ 0, & \text{otherwise.} \end{cases}$$

This formulation describes an eigenvalue problem. Matrix A is a stochastic matrix, with principal eigenvalue of 1. One can use the simple *power method* for solving this problem. To incorporate sequence similarities into the formulation, the normalized node similarity matrix (calculated from an all-to-all BLAST bit scores) can be used. The corresponding modified eigenproblem is as follows:

$$R = \alpha AR + (1 - \alpha)E, \tag{5.12}$$

where E is the normalized node similarity matrix and parameter α represents the tradeoff between network and node similarity. Equation (5.12) is the base equation for all IsoRank-based algorithms.

The approach to learning the weight function is based on the definition of *loss function* $\mathscr{L}$, defined as $\mathscr{L} : \mathscr{A} * \mathscr{A} \to R^+$, which measures the distance of a given alignment from the gold standard alignment used for training. Intuitively, the learned weight vector should assign higher scores to alignments with smaller loss function values, and a score of zero for the correct alignment. The loss function grows as alignments diverge from the correct alignment.

To learn the weight function, Flannick et al. [27] use KEGG Ortholog (KO) groups [28] as the training set. Each training sample contains networks from a set of species, $G^{(i)} = G_1^{(i)}, \ldots, G_n^{(i)}$, with nodes that do not have a KO group removed; the correct alignment $a^{(i)}$ contains an equivalence class for each KO group. Let $[x]_{a^{(i)}}$ denote the equivalence class of $x \in V^{(i)} = \cup_j V_j^{(i)}$ in $a^{(i)}$, and $[x]_a$ denote the equivalence class of protein x under a. One possible definition for the loss function is as follows:

$$\mathscr{L}(a^{(i)}, a) = \sum_{x \in V^{(i)}} |[x]_a \backslash [x]_{a^{(i)}}|. \tag{5.13}$$

Here, $A \backslash B$ represents the set difference between sets A and B. It counts the number of nodes aligned in a that are not aligned in the correct alignment $a^{(i)}$. To learn the weight function, the parameter learning problem is posed as the maximum margin structured learning problem. Given a training set and the loss function, the learned weight function, w, should score each training alignment $a^{(i)}$ higher than all other alignments a by at least $\mathscr{L}(a^{(i)}, a)$. Formally we have the following definition:

$$\forall i, a \in \mathscr{A}^{(i)}, \mathbf{w}.\mathbf{f}(a) + \mathscr{L}(a^{(i)}, a) \leq \mathbf{w}.\mathbf{f}(a^{(i)}), \tag{5.14}$$

where $\mathscr{A}^{(i)}$ is the set of all possible alignments of $G^{(i)}$. The optimal weight function w is then computed using a subgradient descent method.

After finding the optimum w, Graemlin2.0 uses a hill-climbing method for approximating the global alignment of input networks. It starts from an initial alignment containing every node in a separate equivalence class. It then iteratively updates the alignment by evaluating a series of local movements on vertices, computing the alignment score before and after the move, and performing the move that increases the score the most. There are four possible moves for each vertex under consideration:

- Do nothing
- Create a new equivalence class containing only that node
- Move the node to another equivalence class
- Merge the container equivalence class of that node with another equivalence class

This process terminates when an iteration does not increase the alignment score. Graemlin2.0 is available for download at http://graemlin.stanford.edu/graemlin-2.01.tar.gz Datasets needed for training and testing Graemlin2.0 can be downloaded from http://graemlin.stanford.edu/graemlin-2.0_test_files.tar.gz.

3.2.4 Methods Based on Integer Quadratic Programming Formulations

The global alignment problem can be explicitly posed as an *integer quadratic programming (IQP)* problem. Several approaches take this view to the global alignment problem and aim to solve this problem. Before we introduce the IQP formulation of the one-to-one global network alignment problem, we note that the pairwise alignment of a pair of input networks, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, can be formulated as a bipartite graph matching problem. We construct a bipartite graph as follows: let the vertices of the first part of the bipartite graph consist of the vertices in V_1, and the vertices in the second part consist of the vertices in V_2. Connect each node in the first part to every node in the second part, but not to any of the vertices within the same partition. Formally, let us denote the bipartite graph as $G_{Bi} = (V_{Bi}, E_{Bi})$, in which $V_{Bi} = \{V_1 \cup V_2\}$ and $E_{Bi} = \{(v_i, v_j) \in V_1 * V_2\}$. This graph is a complete bipartite graph, represented as $K_{m,n}$. We claim that every one-to-one network alignment between G_1 and G_2, is equivalent to a matching in the constructed bipartite graph, since any one-to-one network alignment assumes that each vertex in first network is mapped to at most one node in the second network, and correspondingly any matching in the bipartite graph, is a subset of E_{Bi} such that no two edges share the same endpoint. This is equivalent to the condition that each node in the first graph should be aligned with at most one node in the second graph.

Following this bijection, one can extend the concept of matching to *maximum weight matching*, to find an *optimal* one-to-one global network alignment. Having set the *appropriate* edge weights in the bipartite graph, one may argue that the maximum weight bipartite matching (which can be found using the Hungarian algorithm [24] in $O(\max\{m,n\}^3)$ time), is equivalent to the optimal one-to-one network alignment. The pairwise IsoRank algorithm (see Sect. 3.2.2) is an example of this class of problems – it defines the similarity score between nodes in input networks, namely the R matrix, in a way that captures both the node-based and topological similarities around each node, and uses this matrix to weight the edges in the bipartite graph to find the alignment. Integer quadratic programming, on the other hand, aims at explicitly finding the optimal matching and updating the maximum scores of the alignments, in a way that maximizes the node similarity score between matched nodes, as well as the conserved edges in a pair of networks.

Definition 5.3. Integer Quadratic Programming Formulation Given a pair of unweighted graphs, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, represented by their corresponding adjacency matrices $A = (a_{ij})_{m*m}$ and $B = (b_{ij})_{n*n}$, respectively, let the matching variable x_{ij} be equal to one, if node $v_i \in V_1$ is matched to node $v_j \in V_2$. The global network alignment can be formulated as an integer quadratic program as follows:

$$\begin{aligned}\text{Maximize}_X\{\phi(G_1,G_2)\} = {} & \lambda \sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij}x_{ij} + (1-\lambda) \\ & \times \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{m}\sum_{l=1}^{n} a_{ik}b_{jl}x_{ij}x_{kl} \end{aligned} \qquad (5.15)$$

$$\text{Subject to} \quad \begin{cases} \sum_{j=1}^{n} x_{ij} \le 1, & \forall i \in \{1,\ldots,m\}; \\ \sum_{i=1}^{m} x_{ij} \le 1, & \forall j \in \{1,\ldots,n\}; \\ x_{ij} \in \{0,1\}, & \forall i \in \{1,\ldots,m\} \text{ and } \forall j \in \{1,\ldots,n\}. \end{cases}$$

Here, parameter λ adjusts the relative importance of node similarity and edge conservation. The first two constraints ensure that every node in each partition is mapped to at most one node in the other partition, while the last constraint is the integer constraint for variables x_{ij}. This formulation can also be expressed in closed matrix form. Denoting the matching variables x_{ij} and node similarity scores s_{ij} using matrices X and S, respectively, the above definition can rewritten as:

$$\text{Maximize}_X\{\phi(G_1,G_2)\} = \lambda XS + (1-\lambda)AXB^T \bullet X \tag{5.16}$$

$$\text{Subject to} \quad \begin{cases} X1_m \leq 1_n, X^T 1_n \leq 1_m & \text{Matching constraints;} \\ x_{ij} \in \{0,1\}, & \text{Integer constraint.} \end{cases}$$

Here, $\bullet$ is the inner-product operator between matrices, and 1_m and 1_n are vectors of all ones, of sizes m and n, respectively. To generalize the problem to arbitrary bipartite graphs, (E_{Bi} does not correspond to a complete bipartite graph), we formulate the problem differently. Let vector X_υ, of size $|E_{Bi}| = |V_1| * |V_2| = m * n$, denote the vectorization of X, and vector S_υ of the same size denote the vectorization of S. Let matrix C be a matrix of size $|E_{Bi}| * |E_{Bi}|$, in which element $C_{e_1,e_2} = 1$, for any $e_1 = (i_1, j_1), e_2 = (i_2, j_2) \in E_{Bi}$, if $(i_1, i_2) \in E_1$ and $(j_1, j_2) \in E_2$, and zero otherwise. An entry in matrix C indicates whether or not a pair of matchings, $i_1 \to j_1$ and $j_1 \to j_2$, result in a conserved edge in the input networks. Finally, let matrix D, of size $|V_{Bi}| * |E_{Bi}|$, be the unoriented incidence matrix of the bipartite graph. The general IQP can be written as follows:

$$\text{Maximize}_X\{\phi(G_1,G_2)\} = \lambda S_\upsilon^T X + (1-\lambda)X_\upsilon^T C X_\upsilon \tag{5.17}$$

$$\text{Subject to} \ : \ DX_\upsilon \leq \mathbf{1}, x \in (\{0,1\}^{m*n})^T,$$

where $\mathbf{1}$ is the vector of all ones, of size $|V_{Bi}| = |V_1| + |V_2| = m + n$.

Equation (5.15) was initially proposed by Li et al. [29], who showed that the constraints in the formulation have a unimodular property. This implies that the problem can be relaxed to quadratic programming with an integral solution in the general case. Furthermore, they proved the sufficient conditions to ensure that the quadratic programming will have an integer solution.

The associated algorithm, called MNAligner, is used to align PPI networks of yeast (*S. cerevisiae*) and fruit fly (*D. melanogaster*) (from [22]), as well as a pair of metabolic pathways for *E. Coli* and yeast (*S. cerevisiae*). To deal with the computational complexity associated with large networks, Li et al. apply a network clustering algorithm to input network first, and then apply their method to identify conserved regions in the smaller subgraphs. Matlab code for the algorithm is available from http://zhangroup.aporc.org/bioinfo/MNAligner or http://intelligent.eic.osaka-sandai.ac.jp/chenen/software/MNAligner.

The closed form in (5.16) and (5.17) is first introduced by Bayati et al. [30]. They also show that IsoRank is an approximate solution of the integer quadratic programming, that does not explicitly satisfy the constraints. They also propose a modification of the IsoRank formulation by restricting the number of edges in the bipartite graph by eliminating unpromising edges. This makes the algorithm more suitable for large sparse graphs (where the number of nodes in input graphs are in order of hundreds of thousands). Their implementation in Matlab as well as test cases are available for download at http://www.stanford.edu/~dgleich/publications/2009/netalign/.

Klau [31] proposes a similar formulation, albeit with different notation, and a different relaxation technique. In this approach, the IQP is first transformed into an equivalent linear integer program. A relaxation based on the Lagrangian decomposition is then used to solve this problem. The violation of constraints, together with their Lagrange multipliers, are integrated into the objective function. It is known that the solution to the Lagrangian linear program is an upper bound for the linear program, which itself is an upper bound for the network alignment problem. A heuristic is developed for reducing the gap between the fractional upper bound and integer solution. An implementation of this technique is available from https://www.mi.fu-berlin.de/wiki/pub/LiSA/Natalie/natalie-0.9.tgz.

Zaslavskiy et al. [32,33] also use a similar formulation, and propose two different methods for solving it. The first method, called GA, is based on the gradient descent method. GA starts from an initial solution and searches the state space of matchings for an optimal solution based on the gradient of the objective function ϕ. Like all other local search methods, this approach is suitable if we can start from a "good" initial solution that is close enough to the optimal solution. Otherwise, it gets stuck in local minima. The second algorithm, called PATH, is based on two relaxations of (5.17), one concave and one convex, over the set of doubly stochastic matrices. PATH starts by solving the convex relaxation, using the Frank–Wolfe method [34], and then iteratively solves a linear combination of convex and concave relaxation by gradually increasing the weight of the concave relaxation and following the path of the solutions thus created. This algorithm is implemented as part of the *Graph Matching (GraphM)* package. This package aims to collect various graph matching methods in a unified framework, and to organize them in a simple, easily extendible software package. The package is freely available from http://cbio.ensmp.fr/graphm/personal_dir/graphm-0.5.tar.gz.

3.3 Multiple Network Alignment: Complexity and Scalability

Increasing amounts of network data requires methods that scale up from aligning pairs of networks to multiple networks from different species. Existing methods have serious limitations with respect to scalability to large numbers of networks and most rely on heuristics. Trade-offs between computational cost of heuristics and their solution quality remains an open and active area of research.

NetworkBlast, proposed by Sharan et al. [7], is applied to the alignment of up to three networks. While this method is able to align multiple networks theoretically, in practice the running time grows exponentially in the number of species, which limits the number of graphs that can be simultaneously aligned. Kalaev et al. [35] improve the running time of this method from $O(n^k)$ to $O(n2^k)$, where n denotes the number of vertices and k denotes the number of networks. The intuition behind this method is to prevent the creation of alignment graph directly, and to build it implicitly as part of the algorithm. This avoids creation of nodes for every set of potentially orthologous proteins (recall that the size of the alignment graph grows exponentially in k). Note that the resulting algorithm still has exponential running time.

All IsoRank-based methods require a quadratic time complexity in the number of input species, k, multiplied by the running time for computing similarities between a pair of networks using the iterative procedure. IsoRank for aligning multiple graphs, as proposed by Singh et al. [25] (see Sect. 3.2.2), takes pairwise similarity matrices, and applies a greedy method to construct an alignment graph based on them. IsoRank-N [26], on the other hand, uses a spectral clustering mechanism to cluster the nodes in input networks based on the pairwise similarity matrices.

Flannick et al. [2] define equivalence classes for constructing the alignment graph, and are able to mimic the progressive sequence alignment technique to achieve linear runtime dependence in number of graphs. As mentioned in Sect. 3.1.3, this approach initially links species using a phylogenetic tree, and at each step merges the two closest networks to create a single alignment graph. This method has been successfully applied to up to ten microbial networks. Note, however, that this heuristic is sensitive to the quality of the phylogenetic tree used to establish the relationship between species.

3.4 Validation Methods

An important problem associated with validating network alignment algorithms is that assessment of the quality of an alignment is not straightforward. The basic concept underlying comparative network analysis is one of transferring "knowledge" from one species to other. This knowledge can be the functional annotation of proteins, functional modules, disease/phenotype, etc. Consequently, before we can evaluate a method, and its associated knowledge transfer, we need to define a unified framework to describe the knowledge, annotate entities, and transfer it among different species. *Ontologies*, which provide a hierarchical framework of categorized consensus vocabularies, provide facilities for formally describing the knowledge about various biological entities. This set of vocabularies can change from context to context, and even in the same context we might have several different frameworks. The most widely used vocabularies describing protein function are the Gene Ontology (GO) [36], Enzyme Commission (E.C.) [37], and MIPS Functional Catalogue (FunCat) [38].

GO consists of three individual, hierarchical ontologies containing terms that describe molecular function (biochemical activity), biological process (pathway), and cellular component (localization). GO terms associated with protein sequences carry evidence codes that describe the experimental or computational evidence for the annotation. E.C., which is commonly used for annotating enzymes in KEGG pathways, is a four-level hierarchy of enzyme nomenclature, describing biochemical activity. MIPS FunCat is a six-level hierarchical scheme used for genome annotation containing over 1,300 terms in 28 general categories [39]. There are different ontologies for describing disease implicated genes, based on their relation to different disease related pathways. As an example, NetPath [40], at this time contains ten immune and ten cancer signaling pathways. OMIM [41] is a frequently accessed database related to genetic variants associated with phenotypes.

Here, we primarily focus on methods for quality assessment in function prediction using comparative analysis. Knowledge relating to annotations is partial and one is interested in using methods such as network alignment to expand this knowledge. This enhancement is hard to assess, especially since the available knowledge is not reliable or even homogeneous. As an example, GO annotations have different *tags* based on their annotating methodology, and GO annotations tagged as IEA (electronic annotation), ISS (pure sequence-based annotation), or ND (annotation without documented evidence) are known to be unreliable. This heterogeneity and incompleteness in data makes it hard to define measures for evaluating the quality of different methods. Furthermore, cellular entities typically participate in different processes, and thus have multiple annotations. Considering all of the aforementioned limitations, one must consider a gold standard, and evaluate methods based on this gold standard.

Since there have been different methods proposed for evaluating the consistency of functional annotation mappings, we briefly review different approaches. These approaches are based on given mappings between nodes of the input networks. Singh et al. [25] propose the following methodology for computing *functional coherence* as their quality assessment measure: Given an ortholog list, they initially extract equivalence classes that have at least a fraction k of their proteins with at least one GO term, which they set $k = 80\%$ in their multiple alignment method using IsoRank (see Sect. 3.2.2). Next, they collect all of the GO terms corresponding to any protein in each remaining equivalence class (except those with ISS, IEA or ND tags). To compare these GO term lists, they map each GO term into a *standard form*, which they define as subset of GO terms that are at a distance of five from the root of the GO tree, and each GO term t is mapped to its ancestor(s) at this level. In this way, they not only map the annotation to a common level in the GO hierarchy, but also eliminate functional annotations that are not specific enough. Having the set of proteins in each equivalence class annotated with homogenized set of GO terms, they proposed an intra equivalence class scoring, followed by averaging of scores in different classes. To evaluate the functional coherence in each equivalence class, Singh et al. first define the similarity score between any pair of GO terms used to annotate proteins in each equivalence class as follows: let S_i and S_j be the set of proteins in the equivalence class annotated by standardized GO terms t_i and t_j, respectively. The pairwise similarity score between t_i and t_j is defined as:

$$\mathrm{sim}(t_i,t_j) = \frac{S_i \cap S_j}{S_i \cup S_j} \tag{5.18}$$

Note that this similarity score is symmetric, and is bounded by 0 and 1. Next, to find the functional coherence in each equivalence class, they find the median over all possible pairwise combinations of GO terms in each equivalence class. Finally, as mentioned earlier, they average over functional enrichments of all classes.

They propose different methods for evaluating IsoRankN (see Sect. 3.2.2) – *consistency* and *coverage*. The former is defined as the mean entropy of the predicted clusters. More formally, consistency of a given cluster S_v^* is defined as:

$$H(S_v^*) = H(p_1, p_2, \ldots, p_d) = \sum_{i=1}^{d} p_i \log p_i, \tag{5.19}$$

where p_i is the fraction of the proteins in S_v^* with GO or KEGG group ID i. They also propose a normalization of entropy scores by the cluster size as, $H_{\text{norm}}(S_v^*) = \frac{1}{\log d} H(S_v^*)$. The coverage of an alignment method is measured by the number of clusters containing proteins from at least k species, where k is an adjustable parameter. An alternate definition for coverage is proposed by Kalaev et al. [35] based on the enrichment of predicted groups with respect to known ontologies derived from either GO or KEGG.

Flannick et al. [2] propose two different sets of measures to mimic *sensitivity* and *specificity*, respectively. They assess the former by counting the number of KEGG pathways in species that are aligned together correctly, meaning that at least three proteins in each pathway are aligned with their counterparts in the other species. To measure the specificity, they propose two methods. First, to compute the specificity based on GO terms, they assign to each protein all of its annotations from level 8 or deeper in the GO hierarchy, and then calculate the alignment enrichment using GO TermFinder [42]. The alignment is considered enriched, if the p-value of the alignment is less than 0.01. Second, they measure the specificity based on the fraction of nodes that have KEGG orthologs, but are aligned to any nodes other than their KEGG orthologs.

An alternate method for assessing alignment methods, is to measure the number of conserved edges. Conservation in this sense means that a pair of nodes v_i^1 and v_j^1 are aligned to their orthologs v_i^2 and v_j^2, and there is an edge both between v_i^1 and v_j^1, as well as v_i^2 and v_j^2, indicating that the alignment *conserved* those edges.

3.5 Databases

There are a number of databases for comparative network analysis. The first set of sources contain interactomes of different species. One of the most commonly studied interactomes, is the PPI network. The following databases are frequently used for PPI data:

- Biomolecular Interaction Network Database (BIND) [43] is a database of full descriptions of interactions, molecular complexes, and pathways. Development of the BIND 2.0 data model has led to the incorporation of virtually all components of molecular mechanisms, including interactions between any two molecules composed of proteins, nucleic acids, and small molecules. The BIND database can be accessed through http://www.bind.ca/.

- The Database of Interacting Proteins (DIP) [44] catalogs experimentally determined interactions between proteins. It combines information from a variety of sources to create a single, consistent set of PPIs. The data stored in DIP has been curated, both manually, by expert curators, and automatically, using computational approaches that utilize knowledge about the PPI networks extracted from the most reliable, core subset of DIP data. In addition to the interaction information, DIP includes additional data regarding the proteins participating in PPI networks. This database is available on http://dip.doe-mbi.ucla.edu/.
- IntAct [45] provides an open framework for storing, presenting, and analyzing protein interactions. The web interface provides both textual and graphical representations of protein interactions, and allows exploration of interaction networks in the context of the GO annotations of the interacting proteins. A web service allows direct computational access to interaction networks in XML. All IntAct services are accessible through http://www.ebi.ac.uk/intact.
- The Biological General Repository for Interaction Datasets (BioGRID) [46] is another curated database of protein–protein and genetic interactions. It aims to provide a comprehensive resource for protein-protein and genetic interactions for all major model organisms, while attempting to remove redundancy, to create a single mapping of interactions. It can be accessed from http://www.thebiogrid.org/.
- The Molecular INTeraction database (MINT) [47] extracts, curates, and stores experimental information about physical interactions between proteins from previously published results in peer-reviewed journals. This database is accessible from http://mint.bio.uniroma2.it/mint/.
- MPact [48] is a PPI database that is targeted to *yeast (S. cerevisiae)*. The complete dataset, as well as user-defined subnetworks can be retrieved in the PSI-MI format from http://mips.gsf.de/genre/proj/mpact.

DIP, IntAct, BioGRID, MINT, and MPact are participating databases in the International Molecular Exchange Consortium (IMEx), a group of the major public interaction data providers. The databases of IMEx work together to prevent duplications of effort, collecting data from nonoverlapping sources and sharing curated interaction data. There are also several databases related to cellular pathways, which are briefly reviewed here:

- Kyoto Encyclopedia of Genes and Genomes (KEGG) [28], which is publicly available at http://www.genome.ad.jp/kegg/, is a collection of online databases of genomes, enzymatic pathways, and biochemicals. The *pathway* database stores networks of molecular interactions in the cells, and their variants specific to select organisms. They cover different areas of interest including metabolism, genetics, cellular processes, human diseases, and drug development. The database also provides a standardized method for representing pathways that a protein takes part in using the *KEGG Orthology (KO)*.

- BioCyc [49] is a collection of databases publicly available at http://biocyc.org/. Databases within BioCyc describe genome and pathway information for individual organisms. EcoCyc and MetaCyc are the two databases within BioCyc, which are well curated from scientific literature.
- Netpath [40] is a curated resource of human signal transduction pathways, which can be accessed at http://www.netpath.org/. It currently consists of ten immune and ten cancer signaling pathways. These pathways contain information pertaining to PPIs, catalytic reactions, translocation events, and genes that are differentially regulated upon stimulation of receptors by their specific ligands.
- Reactome [50] is a curated, peer-reviewed resource of human biological processes publicly available at http://www.reactome.org/. The largest set of entries refers to human biology, however, it also covers a number of other organisms as well. GO is used to describe the subcellular locations of molecules and reactions, molecular functions, and the larger biological processes that a specific reaction is part of.
- NCI-Nature Pathway Interaction Database (PID) [51] is a free biomedical database of human cellular signaling pathways available at (http://pid.nci.nih.gov/). The database contains information about molecular interactions and reactions that take place in cells, with a specific focus on processes relevant to cancer research and treatment.

In addition to interactome and pathway databases, there are several sequence-related (genes/proteins) databases. Currently, UniProt [52], which is accessible at http://www.uniprot.org/, is the universal protein database. It is a central repository of protein sequences that integrates Swiss-Prot, a reliable database from European Bioinformatics Institute (EBI), and Swiss Institute of Bioinformatics (SIB), TrEMBL, a less reliable database that covers a wider range of proteins, and Protein Sequence Database (PSD), from Protein Information Resource (PIR). Three major databases storing gene sequences are:

- DNA Data Bank of Japan (DDBJ) [53], which is maintained by National Institute of Genetics (NIG) in the Shizuoka prefecture of Japan, which is publicly available at www.ddbj.nig.ac.jp/.
- EMBL Nucleotide Sequence Database [54], which is maintained by the European Bioinformatics Institute (EBI), available at http://www.ebi.ac.uk/embl.
- GenBank [55], maintained by National Center for Biotechnology Information (NCBI) as part of the International Nucleotide Sequence Database Collaboration, or INSDC, available at www.ncbi.nlm.nih.gov/genbank/.

These databases are the main repositories for gene sequences for all organisms, and are members of The International Nucleotide Sequence Database (INSD). They exchange newly submitted gene sequences frequently (daily) minimize inconsistencies. There are also a number of species-specific databases. These databases typically integrate information from different sources to construct a uniform database of all the information specific to a species. Some well-known species-specific databases include:

- Flybase [56], accessible at http://flybase.bio.indiana.edu/, is an online database of the biology and genome of the model organism *fruit fly (D. melanogaster)*. It contains a complete annotation of the *D. melanogaster*. It also includes a searchable bibliography of research on Drosophila genetics.
- The Arabidopsis Information Resource (TAIR) [57] maintains a database of genetic and molecular biology data for the model organism *plant (Arabidopsis thaliana)*, at http://www.arabidopsis.org/. Data available from TAIR includes the complete genome sequence, along with gene structure, gene product information, metabolism, gene expression, DNA and seed stocks, genome maps, genetic, and physical markers.
- Mouse Genome Database (MGD) [58] is an integrated data resource for mouse genetic, genomic, and biological information, at http://www.informatics.jax.org/. MGD includes a variety of data, ranging from gene characterization and genomic structures, to orthologous relationships between mouse genes and those of other mammalian species, to maps (genetic, cytogenetic, physical), descriptions of mutant phenotypes, characteristics of inbred strains, and information about biological reagents such as clones and primers. Data is accessed via search/retrieval Web forms and displayed as tables, text, and graphical maps, with supporting primary data. A rich set of hypertext links is provided, such as those from gene and clone information to DNA and protein sequence databases (GenBank, EMBL, DDBJ, SWISS-PROT), from bibliographic data to PubMed, from phenotypes to Online Mendelian Inheritance in Man (OMIM), and from gene homology records to the genomic databases of other species.
- Rat Genome Database (RGD) [59], accessible at http://rgd.mcw.edu/, stores genomic, genetic, functional, physiological, pathway and disease data for the laboratory rat, as well as comparative data for rat, which is a major model organism for the study of human disease.
- Saccharomyces Genome Database (SGD) [60] stores information about the chromosomal features and gene products of the budding yeast *S. cerevisiae*, which can be publicly accessed at http://www.yeastgenome.org/.

4 Applications

Network alignment has been successfully applied to a variety of problems, including function prediction for unannotated proteins, investigating cellular machinery, comparative analysis of evolutionary events, and integrating biological networks with prior data sources for disease diagnoses.

4.1 Projecting Functional Annotations

While high-throughput methodologies for assessing protein function are emerging, computational methods are essential for complementing these experimental techniques. Evolutionary events and analyses have been shown to be effective in studying biomolecular functions across species. Comparative analyses across evolutionarily close species, such as humans (*H. sapiens*) and mice (*M. musculus*), and model organisms such as yeast (*S. cerevisiae*), nematode worm (*C. elegans*), and fruit fly (*D. melanogaster*) (because of their short life cycle), have provided critical insight into structure and function of various proteins [61].

Understating phylogenetic relationships among proteins can help in predicting their structure and function. Two proteins with similar sequences are known to be *homologous*. If a pair of homologous proteins have evolved from a common ancestor by *speciation* event(s), they are referred to as *orthologs*. Proteins can also be separated by *duplication* event(s) – such proteins are called *paralogs*. Paralogous proteins, contrary to orthologous proteins, can, and usually do, diverge in their function after duplication. They can be classified into two different classes: *in-paralogs* (also known as *recent* paralogs), in which pairs of proteins are duplicated after a speciation event, and *out-paralogs* (also known as *ancient* paralogs), in which the duplicated event precedes the speciation event. In the former case, proteins are more likely to be true functional orthologs, since there is shorter distance between the duplicated ancestor and its descendants.

Early computational methods for predicting protein functions are primarily sequence-based. Sequences of proteins from different species are compared to find homologous proteins. Two examples of sequence-based models are Clusters of Orthologous Groups (COG) [15] and Inparanoid [62]. COG defines functional orthologs using sets of proteins that contain best BLAST matches across a minimum of three species. The Inparanoid approach is a sequence-based method for finding functional annotation. It uses clustering to derive ortholog families, leaving some of the orthology relations ambiguous. When the homology is not ambiguous, especially in cases where the function is essential to the evolutionary fitness, the pair of homologous proteins usually are functional orthologs, carrying the same set of functions. On the other hand, when we have multiple homologous proteins in different organisms, there is an ambiguity about the true functional orthologs, since these may result from different evolutionary events.

An all-versus-all BLAST method for predicting protein functions is often unable to distinguish between out-paralogs and in-paralogs, and thus results in false-positives. Different methods have been proposed to remedy this problem. Comparative network analysis is one of the most promising methods. The motivation behind the use of comparative network analysis is that out-paralogs, which are ancient, had more time to diverge in their patterns of interactions. One may use these differences in interaction patterns to make a decision regarding the elimination of out-paralogous proteins. Most of the global alignment methods discussed in Sect. 3.2, are used to transfer functional annotations based on this hypothesis.

4.2 Conserved Functional Modules Across Species

Biological systems can often be decomposed into smaller subsets, known as modules. Modules are a sets of cohesive entities that are loosely connected to the rest of the system [63]. Hartwell et al. [64] hypothesized that biological processes within individual cells are carried out by such modules, also called "functional modules." These are discrete entities composed of various molecules, whose functions are separable from others, and whose functions are manifested in their interaction patterns.

It has been shown that most of cellular interactome, including PPI, metabolic, and GRNs, have modular structure. Protein complexes in PPI networks [65, 66], metabolic pathways in metabolic networks [67], and signal transduction pathways in GRNs [68, 69], are examples of modules that have specific functions in their corresponding networks. However, both decomposition and functional annotation of the modules pose significant challenges from points of view of model and method development. Comparative network analysis is known to be one of the most powerful methods for decomposing networks of multiple species to extract their common functional modules. These methods are based on the idea that conservation of specific substructures across species implies their functional coherence (for a brief overview of other methods, please see Sect. 5.1).

Most of the local alignment methods discussed in Sect. 3.1 can be used to extract conserved substructures in the networks of multiple species, and to predict functional modules. For example, the use of PathBLast [5] to align protein interaction networks of two distantly related species, yeast (*S. cerevisiae*) and bacterium (*H. pylori*), uncovers remarkable conserved pathways among them. It is also used for aligning protein interaction networks of *P. falciparum* with other eukaryotes (yeast (*S. cerevisiae*), fruit fly (*D. melanogaster*), and worm (*C. elegans*)) and yeast (*S. cerevisiae*), and bacterium (*H. pylori*), to identify their conserved pathways (please see Sect. 3.1.1 for more details). MAWISH [9, 10] and NetworkBlast-M [7] are used to find conserved protein complexes, and to identify relationships among different biological process in distinct organisms. Graemlin1.0 [2] is used for transferring functional annotations of known modules in one species to another. It used two approaches – the first one transfers functional annotation from a protein, to other aligned proteins whose annotations are unknown. This is similar to sequence based methods, however, network-based methods have been shown to be more accurate since they rely on both sequence and interactions. The second approach is specific to network alignment – it works on the thesis that the function of landmark proteins is shared by other proteins in the alignment.

4.3 Studying Evolutionary Events

Evolution manifests itself in variations, for example, mutations on the genome that impact structure and function of associated proteins [68]. These changes in turn affect protein interactions, metabolic reactions, and genetic interactions [64]. Despite the key roles of these interaction networks in structural and functional characterization, study of evolutionary trajectories remains an active area of investigation. The evolution of protein interaction networks depends on the modification of genes that produce proteins and the way the general structure of the network has been impacted over the evolutionary history [71]. Phylogenetic analysis of protein interaction networks is most commonly performed through network comparison, based on the idea that a common ancestor is shared by all distinct organisms [1].

Network alignment can be used in different levels to uncover evolutionary relations. At lowest level, each protein can be represented by its tertiary structure graph, and the corresponding structures can be compared using network alignment. As mentioned earlier, evolution affects cellular process by mutations on nucleotide sequences. These result in changes in amino acid sequences and consequently in their structure and function. Conversely, identifying similarities and differences in protein structures can uncover their evolutionary history [72]. Comparative network analysis can also be used to overcome the ambiguity problem among homolog proteins that pure sequence-based methods often fail to do. As discussed in Sect. 4.1, discriminating in-paralogs and out-paralogs, which is an important component of finding the order of duplication/speciation events, can be efficiently performed using network alignment of protein interactions of different species.

Comparative analysis can also be applied at a modular level to help understand evolutionary events. For example, evolutionary biologists have extensively studied genes related to aging, and on understanding mechanisms leading to cell death. The role of biological networks in explaining the complex traits provides exciting new avenues to extend current efforts focused on the study of individual genes. Promislow [73] examined the subset of yeast proteins related to senescence, and collates them with subsets of other traits. He found that proteins (and corresponding genes) related to senescence have higher connectivity compared to other proteins; four of five examined traits were unrelated to senescence, and they did not have notable connectivity in comparison to those related to senescence. His final conclusion was that genes associated with aging produce proteins that are more highly connected and have greater pleiotropy (are associated with multiple phenotypes that seem completely independent from each other). Wagner et al. [74] illustrated that the rate of evolution in highly connected nodes between modules is significantly higher than the nodes found in a module. Finally, Yosef et al. [75] developed a framework for investigating the evolutionary trajectory of protein complexes, besides the role of the self-interacting protein's duplication in complex evolution. In this study, phylogenetic analysis is used to age the proteins in each complex, which were identified by network alignment using NetworkBLAST (see Sect. 3.1.1) or network clustering using MCL algorithms. They also investigate whether the members of a protein complex evolve together. Their results show that complex proteins emerge early in evolution, and evolve together over history.

Network alignment can also be used as a compare function between the interactomes of different species. This in turn can be interpreted as the distance in their cellular structure. From this point of view, network alignment can be used as the pairwise distance of a set of species, to link them and reconstruct the phylogenetic tree. Most of the previously mention methods, for example, IsoRank and IQP-based methods (see Sect. 3.2), align the entire interactome as a single object, and give a unique alignment score to each pair of networks. However, due to the computational cost of aligning large networks, all of these algorithms must deal with intractable graph comparisons, besides the high degree of noise in interaction data. Towfic et al. [76] propose an algorithm for aligning large biological networks based on the alignment of their subgraphs, which are scored by a graph kernel. The computed score of these alignments can be used for clustering species and recovering phylogenetic information. Erten et al. [70] alleviate these difficulties by designing a slightly different approach, named MOPHY, to extract evolutionary trajectory based on conserved modules. The fundamental idea behind MOPHY that cohesive interaction patterns have strong tendency to be conserved over evolutionary history. MOPHY initially identifies the modular subgraphs in different networks independently. It then maps these modules to networks of other species to understand the conservation and divergence of different modular processes in these networks. Finally, it uses these modules to compute overall phylogeny of the networks by comparing the module maps together using feature vectors. Results from this algorithm show that modularity based analysis can be used to gain deeper insight into functional evolution, and phylogenetic analysis of individual module.

4.4 Disease Discovery

The availability of complete genomes, proteomic, and metabolic data combined with phenotype characterization provide significant new avenues for understanding and treating disease [77]. It has been shown that the function of a gene in disease depends on the locus of its protein in protein interaction networks [78]. These intricate relationships in cellular networks establish the role of network analysis.

Different classes of human diseases such as cancer types, autoimmune disorders, hormone diseases, genetic disorders, infectious diseases, neurological disorders,and mental illnesses are caused by defects in genetic structure or cellular metabolism. This can be the result of the pathogen's infection or genetic variations such as missing, mutation, or extra copy of a gene. Understanding the genetic makeup of diseases is the initial step toward analyzing different diseases and intervention.

A variety of biochemical networks have been shown to conform to a scale-free structure. While robustness is one of the key attributes of scale-free networks, targeted variations at specific loci and positions lead to dysfunctional behavior. In fact, this characteristic of biological networks supports the observation that several mutations must occur for the onset of a disease like cancer [80]. Studying the structural changes to networks caused by diseases, is an essential component of understanding underlying mechanisms, and consequently their cure.

Many of the methods described in this chapter are directly applicable to disease discovery and characterization as well. As shown earlier, function prediction is carried out by first annotating proteins in different networks that have known GO or KEGG annotations, and then transferring (projecting) these annotations to their putative orthologs after aligning networks. Similarly, one may annotate proteins, genes, or other cell components in the networks by their known relations to different diseases, and transfer this knowledge to putative orthologs to predict disease implicated genes. In a similar fashion, functional module discovery is also readily extendable to disease discovery – one may annotate pathways, complexes, and disease related modules in general, and transfer this knowledge to the aligned substructures. Finally, network-based phylogenetic studies can be used for understanding disease. Phylogenetic trees are useful in uncovering the evolution of viral strains [81]. Such studies can be used to explain how some viruses, for example, canine (a virus that transfers from cats to dogs), can jump from one species to another. This analysis leads to a better understanding of viruses such as avian influenza that can transfer to humans from other species such as birds or pigs. Using phylogenetic methods to identify the relationships between HRV (rhinovirus) strains, as the pathogen for common cold, may lead to novel therapies, and more effective drugs, by elucidating structure, function, interactions, and context [82]. Investigation of human tumor subtypes using phylogenetic methods leads to identification of differentiation-related genes [83].

Understanding pathogens and uncovering the way they affect the normal cells and turn them to infected cells is a fundamental challenge. Comparative network analysis provides an important tool for such analysis. The single cell parasite (*P. falciparum*), is responsible for one million deaths every year around the world from malaria. One of the key challenges in dealing with malaria is that falciparum becomes resistant to the anti-malarial drugs. Falciparum is a human parasite, therefore it causes disruption of pathways active in falciparum without harming normal functional human pathways. Consequently, pathways that are different between the parasite and the human cell provide promising therapeutic targets. Comparative network analysis can help in revealing conserved pathways between falciparum and other eukaryotes, which implicitly help in finding the conserved pathways of falciparum and human, and assist in drug design.

To find conserved pathways, Suthram et al. [84] aligned the protein interactome of falciparum with the protein interactome of yeast (*S. cerevisiae*), worm (*C. elegans*), fruit fly (*D. melanogaster*), and the bacterial pathogen (*H. pylori*), using the PathBlast algorithm (see Sect. 3.1.1). Results from their study showed that falciparum has just three conserved complexes with yeast and no conserved complex with other species. However, yeast, fly, and worm have significant numbers of conserved complex among each other. While this is preliminary research, it shows that falciparum is significantly different from other model organisms and this poses challenges.

Regulatory enzymes have essential roles in cellular metabolism. Among them, phosphorylation is a key event in regulation. Phosphorylation sites are, however, short and changeable, unlike proteins domains that are conserved over longer periods of time. Investigating conservation of phosphorylation sites by sequence similarity is too hard and inefficient. To overcome this problem, Heng et al. [85] investigated the conservation of protein phosphorylation events at sequence, and networks levels for a set of species – human (*H. sapiens*), yeast (*S. cerevisiae)*, nematode worm (*C. elegans*), and fruit fly (*D. melanogaster*). At sequence level, they found *core sites* by identifying conserved phosphorylation sites that are positionally conserved between human and at least one target species. Among a total of 23,977 human phosphorylation sites found across 6,456 phospho-proteins encoded by 6,293 genes, they identify a subset of 479 core sites that are conserved between human, and at least one target species in 344 proteins encoded by 337 human genes. However, phosphorylation sites are often positioned in disordered regions, which are changeable. Therefore they cannot be used to show evolutionary conservation at sequence level. Hung et al. constructed a kinase-substrate network for target model organisms, and applied a network alignment method to extract the conserved human kinase-substrate network, also known as *core net*. Among a total 25,563 human interactions between 113 kinases and 5,515 substrates, 1,255 interactions between 27 human kinases and 778 substrates encoded by 759 genes were found. In this study, 1,105 interactions (88% of interactions) and 698 substrates were not found in core sites. Finally, they illustrated significant overlap between human genes coding phospho-proteins and cancer-associated genes as well as OMIM genes [41].

There have been several other studies relating heterogeneous networks – phenotypic and genotypic networks, and applying comparative network analysis to investigate their structural similarities. Wu et al. [86] introduced AlignPI, a method that exploits known gene-disease associations by aligning the phenotypic similarity network with the human PPI network. To align these, human disease phenome and interactome are modeled as graphs. In the former, nodes correspond to disease phenotype and the latter is a network of genes with interaction between their proteins products. In addition, these two networks are connected by the interactions between their genes and related phenotype. The link between networks is constructed based on the gene-phenotype relationships. Finally, they used NetworkBlast to identify locally dense regions of the PPI network and their associated disease clusters. They find that there is a conserved gene module in gene interactome for each known disease module in human phenome. In other words, for each set of phenotypically related diseases, there is a set of associated causative genes for these diseases. Another important study in this area is the work of Goh et al. [87]. In this study, the authors investigate three different networks – *gene disease* (network of disease genes with links between the genes that are involved in one disorder), *human disease* (network of human diseases with links between diseases that share a disease gene), and *diseasome* (network of human disease and disease genes with link between disease and its causal gene). They discovered high level relationships among human disease and their related causal genes. Their result indicates that similar disorders are usually caused by similar genes.

5 Related Efforts

In this section, we briefly overview the relationships between network alignment and other well-known computational problems. We first study the relationship between network alignment and other network-related methods for finding functional modules, including network motifs, network clustering, and network querying. We then discuss other problems that have similar principles and formulations as the network alignment problem. This comparison is especially useful since there are extensive studies in different fields that share similar goals.

5.1 Network Alignment and Other Network Analyses Problems

Network alignment, especially local network alignment, is a powerful technique for finding conserved modules across species (see Sect. 4.2). The basic motivating idea behind identification of conserved modules, is that the existence of specific connected subgraphs that recur in a specific network or across networks is consistent with the tenets of evolutionary theory. Each of these subgraphs, defined by a particular pattern of interactions among vertices, may correspond to a molecular machinery in which specific functions are achieved efficiently [88]. In addition to network alignment, there are other methods for finding functional modules. These methods often use alternate definitions for functional modules.

Modules are often defined independent of function and based on frequency of occurrence. These connected vertex induced subgraphs, which occur at significantly higher frequencies in networks (as compared to random networks), are also known as *network motifs*. *Network clustering* is another approach used for identifying functional modules. The network clustering problem is based on coupling graph vertices in *a single network*, such that vertices in each group are tightly coupled with each other, but are loosely coupled with other vertices. The underlying idea is that functionally related proteins interact with each other, thus need to be in the close proximity. Protein complexes in PPI networks are examples of this class of functional modules.

The network alignment methods discussed in this chapter aim to find a functional mapping between the nodes of the networks being compared. An alternate approach to comparative network analysis is to directly compare the topological properties of these networks. In an attempt to understand the topological characteristics of PPI networks, Pržulj [89] used graphlet distributions across multiple species, where a graphlet refers to a small subgraph with a specified topology. Extensive studies on PPI networks of 14 eukaryotic species showed that this approach outperforms standard topological measures in understanding the functional relationships between different PPI networks.

5.2 Network Alignment vs. Graph Matching

There are different formulations of similarity between two labeled graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, with adjacency matrices A and B, respectively. The most restricted definition is graph isomorphism: G_1 and G_2 are called *isomorphic*, if there exists a mapping $\pi : V_1 \rightarrow V_2$ that maps E_1 to E_2. More formally, two graphs are isomorphic if there exists a permutation matrix P such that $B = PAP^T$. This definition is strict, and cannot be used if the number of vertices in the input graphs is not equal. The *subgraph isomorphism problem* and *maximum common subgraph* are two extensions of the graph isomorphism problem to a more general case where the number of vertices is not equal. In the former, we aim to map the smaller graph to a subset of larger graph, while in the latter, we aim to find a pair of maximum subgraphs in input graphs that are isomorphic. We cannot readily use these extensions because biological networks are usually noisy and contain false positive/negative edges. A more flexible definition of graph similarity is needed.

If one can define the similarity between a pair of nodes in V_1 and V_2, the similarity between G_1 and G_2 can be formulated as a *graph matching problem*. Formally, a matching in graph $G = (V, E)$ is defined as a subset of E without any common vertices, which is also known as independent edge set in G. In other words, any subset of edges in E defines a graph matching in G, if every vertex in V has degree 1 (for detailed definition of graph similarity scores and their relationships to graph matching, please refer to [90]).

An alternate formulation is specifically useful in pairwise one-to-one network alignment. To find the graph similarity between graphs G_1 and G_2, one needs to construct a complete bipartite graph $G_{Bi} = (V_{Bi}, E_{Bi})$, in which $V_{Bi} = \{V_1 \cup V_2\}$, and $E_{Bi} = \{(v_i, v_j) : \forall v_i \in V_1, v_j \in V_2\}$. With this bijection, each one-to-one network alignment between the species, can be viewed as a matching in G_{Bi}, since each alignment assumes that every vertex is at most mapped to one other node, and correspondingly any matching in G_{Bi}, which is a subset of E_{Bi}, does not contain any edges that share the same endpoint. There is a two way connection between matchings and one-to-one network alignment: starting from a matching in G_{Bi}, one can construct the corresponding network alignment of input networks and vice versa. To see this, one may observe that each edge in G_{Bi} defines a potential alignment between a pair of nodes in input networks, while the matching constraint enforces these potential alignments to be one-to-one. This implies that each node is aligned to one other node, at most, from each network.

Among all possible matchings in G_{Bi} (i.e., all pairwise network alignments between G_1 and G_2), one is usually interested in an optimal matching. This is usually achieved by appropriately weighting each edge in the bipartite graph G_{Bi} based on node similarities, as well as local topological similarities, and finding the *maximum weight matching* in G_{Bi}. There are both *exact* and *approximate* algorithms for different versions of the graph matching problem. Among these is a well-known exact polynomial algorithm, known as the Hungarian algorithm [24]. This algorithm has time complexity $O(\max\{V_1, V_2\}^3)$.

5.3 Network Alignment vs. Graph Kernels

Kernel methods are often used in pattern discovery. They can be applied to general data types, including sequences and *graphs* to identify general relations, such as clustering, correlation, and classification. The core of any kernel method is a *kernel function* for measuring the similarity of any given pair of input objects, by mapping them into another space, named *feature space*. Computations in this space are typically easier and input data is more separable. This mapping is defined implicitly, by specifying a kernel function $\kappa : X * X \rightarrow \Re$, as the inner product for the feature space. This is defined as $\kappa(x_1,x_2) = <\phi(x_1),\phi(x_2)>$, where $\phi(.)$ is the embedding function. Note that one does not need to know the mapping $\phi(.)$ explicitly. It suffices to be able to evaluate the kernel function, which is often much easier than computing the coordinates of the points explicitly.

Kernel methods provide useful tools in the analysis of biological networks, since they can be applied at various levels. At the lowest level, one is interested in a kernel function k_v over the set of vertices in *a single graph*. Defining such a kernel function can help us in clustering nodes together based on their similarity. One of the frequently studied examples of this class of problems is the prediction of protein functions in a single network by assigning a kernel function to them that captures both the similarity of the node attributes (amino acid sequences) and local network structure. Diffusion kernels, which are based on random walks in the input graph, are among the most well-known methods for assigning node similarities. At the next level, one is interested in defining k_v for vertices in *multiple graphs*. To compute this kernel, the product graph of the input graphs can be constructed, and the kernel function can be defined based on the random walks in the product graph. The IsoRank method (see Sect. 3.2.2) is similar in nature to this class of problems. At the next level, we aim to define *graph kernels*, instead of *vertex kernels*. Graph kernels can be used to find the similarity between networks, and thus they can be used as a tool for clustering different species based on their similarities. Graph kernels also provide powerful methods for reconstructing phylogenetic trees based on a hierarchical clustering.

Acknowledgments The authors thank Ron Pinter (Technion) and Mehmet Koyutürk (Case Western Reserve) for many useful suggestions.

References

1. Ideker, R.S.T.: Modeling cellular machinery through biological network comparison. Nature Biotechnology **24** (2006) 427–433
2. Flannick, J., Novak, A., Srinivasan, B., McAdams, H., Batzoglou, S.: Graemlin: general and robust alignment of multiple large interaction networks. Genome Research **16**(9) (2006) 1169–1181
3. Kuchaiev, O., Milenković, T., Memišević, V., Hayes, W., Pržulj, N.: Topological network alignment uncovers biological function and phylogeny. Journal of The Royal Society Interface (2010)
4. Kelley, B.P., Yuan, B., Lewitter, F., Sharan, R., Stockwell, B.R., Ideker, T.: PathBLAST: a tool for alignment of protein interaction networks. Nucleic Acids Research **32**(Web-Server-Issue) (2004) 83–88

5. Kelley, B.P., Sharan, R., Karp, R.M., Sittler, T., Root, D.E., Stockwell, B.R., Ideker, T.: Conserved pathways within bacteria and yeast as revealed by global protein network alignment. PNAS **100(20)** (2003)
6. Sharan, R., Ideker, T., Kelley, B.P., Shamir, R., Karp, R.M.: Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. Journal of Computational Biology **12**(6) (2005) 835–846
7. Sharan, R., Suthram, S., Kelley, R.M., Kuhn, T., McCuine, S., Uetz, P., Sittler, T., Karp, R.M., Ideker, T.: Conserved patterns of protein interaction in multiple species. Proceedings of the National Academy of Sciences of the United States of America **102**(6) (2005) 1974–1979
8. Bader, J.S., Chaudhuri, A., Rothberg, J.M., Chant, J.: Gaining confidence in high-throughput protein interaction networks. Nature Biotechnology **22**(1) (2003) 78–85
9. Koyutürk, M., Grama, A., Szpankowski, W.: Pairwise local alignment of protein interaction networks guided by models of evolution. In: RECOMB. (2005) 48–65
10. Koyutürk, M., Kim, Y., Topkara, U., Subramaniam, S., Szpankowski, W., Grama, A.: Pairwise alignment of protein interaction networks. Journal of Computational Biology **13(2)** (2006) 182–199
11. Pastor-Satorras, R., Smith, E., Solé, R.V.: Evolving protein interaction networks through gene duplication. J Theo. Bio. **222** (2003) 199–210
12. Vázquez, A., Flammini, A., Maritan, A., Vespignani, A.: Modeling of protein interaction netwokrs. ComPlexUs **1** (2003) 38–44
13. Wagner, A.: How the global structure of protein interaction networks evolves. Proc. R. Soc. Lond. Biol. Sci. **270**(1514) (2003) 457–466
14. Koyutürk, M., Kim, Y., Subramaniam, S., Szpankowski, W., Grama, A.: Detecting conserved interaction patterns in biological networks. Journal of Computational Biology **13**(7) (2006) 1299–1322
15. Tatusov, R., Fedorova, N., Jackson, J., Jacobs, A., Kiryutin, B., Koonin, E., Krylov, D., Mazumder, R., Mekhedov, S., Nikolskaya, A., Rao, B.S., Smirnov, S., Sverdlov, A., Vasudevan, S., Wolf, Y., Yin, J., Natale, D.: The COG database: an updated version includes eukaryotes. BMC Bioinformatics **4**(1) (2003) 41
16. Srinivasan, B.S., Novak, A.F., Flannick, J., Batzoglou, S., McAdams, H.H.: Integrated protein interaction networks for 11 microbes. In: RECOMB. (2006) 1–14
17. Chor, B., Tuller, T.: Biological Networks: Comparison, Conservation, and Evolution via Relative Description Length. Journal of Computational Biology **14**(6) (2007) 817–838
18. Pinter, R.Y., Rokhlenko, O., Yeger-Lotem, E., Ziv-Ukelson, M.: Alignment of metabolic pathways. Bioinformatics **21**(16) (August 2005) 3401–3408
19. Narayanan, M., Karp, R.M.: Comparing protein interaction networks via a graph match-and-split algorithm. Journal of computational biology: a journal of computational molecular cell biology **14**(7) (September 2007) 892–907
20. Bruckner, S., Hüffner, F., Karp, R.M., Shamir, R., Sharan, R.: Topology-free querying of protein interaction networks. Journal of computational biology : a journal of computational molecular cell biology **17**(3) (March 2010) 237–252
21. Banks, E., Nabieva, E., Peterson, R., Singh, M.: NetGrep: fast network schema searches in interactomes. Genome Biology **9**(9) (2008)
22. Bandyopadhyay, S., Sharan, R., Ideker, T.: Systematic identification of functional orthologs based on protein network comparison. Genome Research **16** (2006) 428–435
23. SinghF, R., Xu, J., Berger, B.: Pairwise global alignment of protein interaction networks by matching neighborhood topology. In: RECOMB. (2007) 16–31
24. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistic Quarterly **2** (1955) 83–97
25. Singh, R., Xu, J., Berger, B.: Global alignment of multiple protein interaction networks. In: Pacific Symposium on Biocomputing. (2008) 303–314
26. Liao, C.S., Lu, K., Baym, M., Singh, R., Berger, B.: IsoRankN: spectral methods for global alignment of multiple protein networks. Bioinformatics **25**(12) (2009) i253–i258

27. Flannick, J., Novak, A.F., Do, C.B., Srinivasan, B.S., Batzoglou, S.: Automatic parameter learning for multiple network alignment. In: RECOMB. (2008) 214–231
28. Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y., Hattori, M.: The KEGG resource for deciphering the genome. Nucleic acids research **32**(Database issue) (2004) D277–280
29. Zhenping, L., Zhang, S., Wang, Y., Zhang, X.S., Chen, L.: Alignment of molecular networks by integer quadratic programming. Bioinformatics **23**(13) (2007) 1631–1639
30. Bayati, M., Gerritsen, M., Gleich, D., Saberi, A., Wang, Y.: Algorithms for large, sparse network alignment problems. In: ICDM. (2009) 705–710
31. Klau, G.W.: A new graph-based method for pairwise global network alignment. BMC Bioinformatics **10**(S-1) (2009)
32. Zaslavskiy, M., Bach, F.R., Vert, J.P.: Global alignment of protein-protein interaction networks by graph matching methods. Bioinformatics **25**(12) (2009) i259–1267
33. Zaslavskiy, M., Bach, F., Vert, J.P.: A path following algorithm for the graph matching problem. IEEE Trans. Pattern Anal. Mach. Intell. **31**(12) (2009) 2227–2242
34. Frank, M., Wolfe, P.: An algorithm for quadratic programming. Naval Research Logistics Quarterly **3** (1956) 95–110
35. Kalaev, M., Bafna, V., Sharan, R.: Fast and accurate alignment of multiple protein networks. In: RECOMB. (2008) 246–256
36. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.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., Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. the gene ontology consortium. Nature genetics **25**(1) (2000) 25–29
37. Tipton, K.F., Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB). Enzyme nomenclature. Recommendations 1992. Supplement: corrections and additions. European journal of biochemistry/FEBS, **223**(1) (1994) 1–5
38. Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., Tetko, I., Güldener, U., Mannhaupt, G., Münsterkötter, M., Mewes, H.W.: The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. Nucleic Acids Res **32**(18) (2004) 5539–5545
39. Hawkins, T., Kihara, D.: Function prediction of uncharacterized proteins. J. Bioinformatics and Computational Biology **5**(1) (2007) 1–30
40. Kandasamy, K., Mohan, S.S., Raju, R., Keerthikumar, S., Kumar, G.S.S., Venugopal, A.K., Telikicherla, D., Navarro, J.D., Mathivanan, S., Pecquet, C., Gollapudi, S.K.K., Tattikota, S.G.G., Mohan, S., Padhukasahasram, H., Subbannayya, Y., Goel, R., Jacob, H.K.K., Zhong, J., Sekhar, R., Nanjappa, V., Balakrishnan, L., Subbaiah, R., Ramachandra, Y., Rahiman, B.A., Prasad, T.K.K., Lin, J.X.X., Houtman, J.C.C., Desiderio, S., Renauld, J.C.C., Constantinescu, S.N., Ohara, O., Hirano, T., Kubo, M., Singh, S., Khatri, P., Draghici, S., Bader, G.D., Sander, C., Leonard, W.J., Pandey, A.: NetPath: a public resource of curated signal transduction pathways. Genome biology **11**(1) (2010)
41. Hamosh, A., Scott, A.F., Amberger, J.S., Bocchini, C.A., Mckusick, V.A.: Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic acids research **33**(Database issue) (2005)
42. Boyle, E.I., Weng, S., Gollub, J., Jin, H., Botstein, D., Cherry, J.M., Sherlock, G.: GO: TermFinder-open source software for accessing gene ontology information and finding significantly enriched gene ontology terms associated with a list of genes. Bioinformatics **20**(18) 3710+
43. Bader, G.D., Donaldson, I., Wolting, C., Ouellette, B.F.F., Pawson, T., Hogue, C.W.V.: BIND–The Biomolecular Interaction Network Database. Nucl. Acids Res. **29**(1) (2001) 242–245
44. Xenarios, I., Rice, D.W., Salwinski, L., Baron, M.K., Marcotte, E.M., Eisenberg, D.: DIP: The Database of Interacting Proteins. Nucleic acids research **28**(1) (2000) 289–291
45. Hermjakob, H., Montecchi-Palazzi, L., Lewington, C., Mudali, S., Kerrien, S., Orchard, S.E., Vingron, M., Roechert, B., Roepstorff, P., Valencia, A., Margalit, H., Armstrong, J., Bairoch, A., Cesareni, G., Sherman, D.J., Apweiler, R.: IntAct: an open source molecular interaction database. Nucleic Acids Research **32**(Database-Issue) (2004) 452–455

46. Stark, C., Breitkreutz, B.J., Reguly, T., Boucher, L., Breitkreutz, A., Tyers, M.: BioGRID: a general repository for interaction datasets. Nucl. Acids Res. **34** (2006) D535–539
47. Chatr-aryamontri, A., Ceol, A., Palazzi, L.M.M., Nardelli, G., Schneider, M.V.V., Castagnoli, L., Cesareni, G.: MINT: the Molecular INTeraction database. Nucleic acids research **35**(Database issue) (2007) D572–574
48. Güldener, U., Münsterkötter, M., Oesterheld, M., Pagel, P., Ruepp, A., Mewes, H.W., Stümpflen, V.: MPact: the MIPS protein interaction resource on yeast. Nucleic Acids Research **34**(Database-Issue) (2006) 436–441
49. Karp, P.D., O.C.M.K.C.G.L.K.: Expansion of the biocyc collection of pathway/genome databases to 160 genomes. Nucleic Acids Research **33** (2005) 6083–9
50. Joshi-Tope, G., Gillespie, M., Vastrik, I., D'Eustachio, P., Schmidt, E., de Bono, B., Jassal, B., Gopinath, G.R., Wu, G.R., Matthews, L., Lewis, S., Birney, E., Stein, L.: Reactome: a knowledgebase of biological pathways. Nucleic acids research **33**(Database issue) (2005) D428–432
51. Schaefer, C.F.F., Anthony, K., Krupa, S., Buchoff, J., Day, M., Hannay, T., Buetow, K.H.: PID: the pathway interaction database. Nucleic Acids Research **37**(Database issue) (2009) 674–679
52. 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.L.: The Universal Protein Resource (UniProt). Nucleic Acids Research **33**(Database-Issue) (2005) 154–159
53. Tateno, Y., Imanishi, T., Miyazaki, S., Fukami-Kobayashi, K., Saitou, N., Sugawara, H., Gojobori, T.: DNA Data Bank of Japan (DDBJ) for genome scale research in life science. Nucleic Acids Research **30**(1) (2002) 27–30
54. Stoesser, G., Tuli, M.A., Lopez, R., Sterk, P.: The EMBL Nucleotide Sequence Database. Nucl. Acids Res. **27**(1) (1999) 18–24
55. Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Sayers, E.W.: GenBank. Nucleic Acids Research **37**(Database-Issue) (2009) 26–31
56. Gelbart, W.M., Crosby, M.A., Matthews, B., Rindone, W.P., Chillemi, J., Twombly, S.R., Emmert, D., Ashburner, M., Drysdale, R.A., Whitfield, E., Millburn, G.H., de Grey, A., Kaufman, T., Matthews, K., Gilbert, D., Strelets, V.B., Tolstoshev, C.: FlyBase: a Drosophila database. The FlyBase consortium. Nucleic Acids Research **25**(1) (1997) 63–66
57. Rhee, S.Y., Beavis, W.D., Berardini, T.Z., Chen, G., Dixon, D.A., Doyle, A., Garcia-Hernandez, M., Huala, E., Lander, G., Montoya, M., Miller, N., Mueller, L.A., Mundodi, S., Reiser, L., Tacklind, J., Weems, D.C., Wu, Y., Xu, I., Yoo, D., Yoon, J., Zhang, P.: The Arabidopsis Information Resource (TAIR): a model organism database providing a centralized, curated gateway to Arabidopsis biology, research materials and community. Nucleic Acids Research **31**(1) (2003) 224–228
58. Bult, C.J., Blake, J.A., Richardson, J.E., Kadin, J.A., Eppig, J.T.: The Mouse Genome Database (MGD): integrating biology with the genome. Nucleic Acids Research **32**(Database-Issue) (2004) 476–481
59. Twigger, S.N., Lu, J., Shimoyama, M., Chen, D., Pasko, D., Long, H., Ginster, J., Chen, C.F., Nigam, R., Kwitek, A.E., Eppig, J.T., Maltais, L., Maglott, D.R., Schuler, G.D., Jacob, H.J., Tonellato, P.J.: Rat Genome Database (RGD): mapping disease onto the genome. Nucleic Acids Research **30**(1) (2002) 125–128
60. Hong, E.L., Balakrishnan, R., Dong, Q., Christie, K.R., Park, J., Binkley, G., Costanzo, M.C., Dwight, S.S., Engel, S.R., Fisk, D.G., Hirschman, J.E., Hitz, B.C., Krieger, C.J., Livstone, M.S., Miyasato, S.R., Nash, R.S., Oughtred, R., Skrzypek, M.S., Weng, S., Wong, E.D., Zhu, K.K., Dolinski, K., Botstein, D., Cherry, J.M.: Gene Ontology annotations at SGD: new data sources and annotation methods. Nucleic Acids Research **36**(Database-Issue) (2008) 577–581
61. Dolinski, K., Botstein, D.: Orthology and functional conservation in eukaryotes. Annual Review of Genetics **41**(1) (2007) 465–507
62. O'Brien, K.P., Remm, M., Sonnhammer, E.L.L.: Inparanoid: a comprehensive database of eukaryotic orthologs. Nucl. Acids Res. **33** (2005) D476–480

63. Pereira-Leal, J.B., Levy, E.D., Teichmann, S.A.: The origins and evolution of functional modules: lessons from protein complexes. Phil. Trans. R. Soc. **361**(1467) (2006) 507–517
64. Hartwell, L.H., Hopfield, J.J., Leibler, S., Murray, A.W.: From molecular to modular cell biology. Nature **402** (1999) C47–C51
65. Spirin, V., Mirny, L.A.: Protein complexes and functional modules in molecular networks. Proc Natl Acad Sci U S A **100**(21) 12123–8+
66. Pereira-Leal, J., Enright, A., Ouzounis, C.: Detection of functional modules from protein interaction networks. Proteins **54** (2004) 49–57
67. Ravasz, E., Somera, A.L., Mongru, D.A., Oltvai, Z.N., Barabasi, A.L.: Hierarchical organization of modularity in metabolic networks. Science **297**(5586) (2002) 1551–1555
68. Ihmels, J., Friedlander, G., Bergmann, S., Sarig, O., Ziv, Y., Barkai, N.: Revealing modular organization in the yeast transcriptional network. Nat Genet **31**(4) (2002) 370–377
69. Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., Friedman, N.: Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nat Genet **34**(2) (2003) 166–176
70. Erten, S., Li, X., Bebek, G., Li, J., Koyutürk, M.: Phylogenetic analysis of modularity in protein interaction networks. BMC Bioinformatics **10**(1) (2009)
71. Robertson, D.L., Lovell, S.C.: Evolution in protein interaction networks: co-evolution, rewiring and the role of duplication. Biochemical Society transactions **37** (2009) 768–771
72. Huan, J., Bandyopadhyay, D., Wang, W., Snoeyink, J., Prins, J., Tropsha, A.: Comparing graph representations of protein structure for mining family-specific residue-based packing motifs. Journal of Computational Biology **12** (2005) 657–671
73. Promislow, D.E.: Protein networks, pleiotropy and the evolution of senescence. In: Proc Biol Sci. Volume 271. (2004) 1225–1234
74. Wagner, G.P., Pavlicev, M., Cheverud, J.M.: The road to modularity. Nat Rev Genet **8**(12) (2007) 921–31
75. Yosef, N., Kupiec, M., Ruppin, E., Sharan, R.: A complex-centric view of protein network evolution. Nucl. Acids Res. **37**(12) (2009)
76. Towfic, F., Greenlee, M.H.W., Honavar, V.: Aligning biomolecular networks using modular graph kernels. In: WABI. Volume 5724 of Lecture Notes in Computer Science, Springer (2009) 345–361
77. Joseph, L., Isaac, K., Albert-Laszlo, B.: Human disease classification in the postgenomic era: a complex systems approach to human pathobiology. Molecular Systems Biology **3**(124) (2007)
78. Chavali, S., Barrenas, F., Kanduri, K., Benson, M.: Network properties of human disease genes with pleiotropic effects. BMC Systems Biology **4**(1) (2010)
79. Lin, Z.: Bioinformatics Basics: Applications in Biological Science and Medicine. Edited by Lukas K. Buehler and Hooman H. Rashidi. Brief Bioinform **9**(3) (2008) 256–257
80. Zhu, X., Gerstein, M., Snyder, M.: Getting connected: analysis and principles of biological networks. Genes and Development **21**(9) (2007) 1010–1024
81. Lei, G., Ji, Q.: Whole genome molecular phylogeny of large dsdna viruses using composition vector method. BMC Evol Bio **7**(41) (2007)
82. Palmenberg, A.C., Spiro, D., Kuzmickas, R., Wang, S., Djikeng, A., Rathe, J.A., Fraser-Liggett, C.M., Liggett, S.B.: Sequencing and analyses of all known human rhinovirus genomes reveals structure and evolution. Science (2009)
83. Riester, M., Stephan-Otto Attolini, C., Downey, R.J., Singer, S., Michor, F.: A differentiation-based phylogeny of cancer subtypes. PLoS Comput Biol **6**(5) (2010)
84. Suthram S, Sittler T, I.T.: The Plasmodium protein network diverges from those of other eukaryotes. Nature (2005)
85. Tan, C.S.H.S., Bodenmiller, B., Pasculescu, A., Jovanovic, M., Hengartner, M.O., Jørgensen, C., Bader, G.D., Aebersold, R., Pawson, T., Linding, R.: Comparative analysis reveals conserved protein phosphorylation networks implicated in multiple diseases. Science signaling **2**(81) (2009)
86. Wu, X., Liu, Q., Jiang, R.: Align human interactome with phenome to identify causative genes and networks underlying disease families. Bioinformatics **25**(1) (2009)

87. Goh, K.I., Cusick, M.E., Valle, D., Childs, B., Vidal, M., Barabási, A.L.: The human disease network. PNAS **104**(21) (2007) 8685–8690
88. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks. Science (New York, N.Y.) **298**(5594) (2002) 824–827
89. Pržulj, N.: Biological network comparison using graphlet degree distribution. Bioinformatics **26** (March 2010) 853–854
90. Zager, L.: Graph similarity and matching. Master's thesis, MIT (2005)

Chapter 6
Pattern Mining Across Many Massive Biological Networks

Wenyuan Li, Haiyan Hu, Yu Huang, Haifeng Li, Michael R. Mehan, Juan Nunez-Iglesias, Min Xu, Xifeng Yan, and Xianghong Jasmine Zhou

Abstract The rapid accumulation of biological network data is creating an urgent need for computational methods on integrative network analysis. Thus far, most such methods focused on the analysis of single biological networks. This chapter discusses a suite of methods we developed to mine patterns across many biological networks. Such patterns include frequent dense subgraphs, frequent dense vertex sets, generic frequent patterns, and differential subgraph patterns. Using the identified network patterns, we systematically perform gene functional annotation, regulatory network reconstruction, and genome to phenome mapping. Finally, tensor computation of multiple weighted biological networks, which filled a gap of integrative network biology, is discussed.

1 Introduction

The advancement of high-throughput technology has resulted in the rapid accumulation of data on biological networks. Coexpression networks, protein interaction networks, metabolic networks, genetic interaction networks, and transcription regulatory networks are continuously being generated for a wide-range of organisms under various conditions. This wealth of data represents a great opportunity, to the extent that network biology is rapidly emerging as a discipline in its own right [7, 40]. Thus far, most of the computational methods developed in this field have focused on the analysis of individual biological networks. In many cases, however, a single network is insufficient to discover patterns with multiple facets and subtle signals. There is an urgent need for methods supporting the integrative analysis of *multiple* biological networks.

X.J. Zhou (✉)
Program in Computational Biology, Department of Biological Sciences,
University of Southern California, Los Angeles, CA 90089, USA
e-mail: xjzhou@usc.edu

M. Koyutürk et al. (eds.), *Functional Coherence of Molecular Networks in Bioinformatics*, DOI 10.1007/978-1-4614-0320-3_6,
© Springer Science+Business Media, LLC 2012

Biological networks can be classified into two categories: (1) physical networks, which represent physical interactions among molecules, for example, protein–protein interaction, protein–DNA interaction and metabolic reactions; and (2) conceptual networks, which represent functional associations of molecules derived from genomic data, for example, coexpression relationships extracted from microarray data and genetic interactions obtained from synthetic lethality experiments. While physical networks are still limited in size, the large amount of microarray data allows us to infer conceptual functional associations of genes under various conditions for many model organisms, thus providing a great deal of valuable information for studying the functions and dynamics of biological systems. Although the methods and experiments described in this chapter are applicable to any type of genome-wide network, we use coexpression networks throughout the chapter due to their abundant availability. We transform each microarray dataset into a coexpression network, where nodes represent genes and the edges can be either weighted or unweighted. In a weighted coexpression network, the edge weights can be coexpression correlations; in an unweighted network, two genes are connected with an edge only if their coexpression correlation is higher than a given threshold. Given k microarray datasets, we can construct k networks with the same node set but different edge sets. We refer to this arrangement as a *relation graph set*, since each network provides information on different relationships among the same set of vertices. Note that in a coexpression network, each gene occurs once and only once. The coexpression networks, therefore, have distinct node labels, and we avoid the NP-hard "subgraph isomorphism problem." We also note that our study is distinct from the body of work on comparing biological networks across species [25, 28–30, 42], where the nodes in different networks can have a many-to-many mapping relationship. The methods described here focus on comparing networks from the same species but generated under different conditions.

This chapter describes several types of patterns that can only be discovered by analyzing multiple graphs, and a set of computational methods designed for mining these patterns. First, we discuss algorithms to identify recurrent patterns in multiple unweighted networks. Next, we define and mine differential patterns in multiple unweighted networks. Finally, we introduce an advanced mathematical model suitable for analyzing multiple weighted networks. We will also show how to use the identified patterns to perform gene function prediction, transcription module reconstruction, and transcriptome to phenome mapping.

2 Mining Recurrent Patterns in Multiple Networks

On account of the noisy nature of high-throughput data, biological networks contain many spurious edges which may lead to the discovery of false patterns. However, since biological modules are active across multiple conditions, we can easily filter out spurious edges by looking for patterns that occur in multiple biological networks. For example, we have demonstrated experimentally that recurrent dense subgraphs in multiple coexpression networks often represent transcriptional and

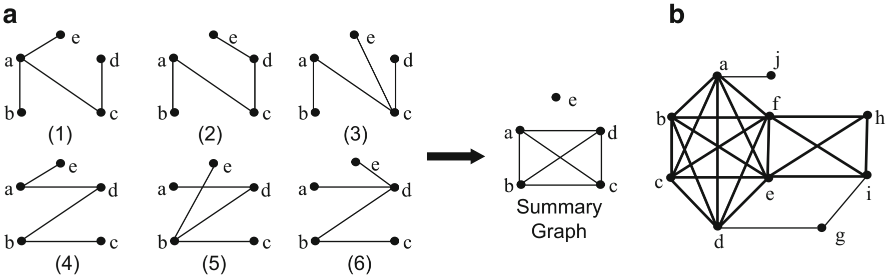


Fig. 6.1 (**a**) Given six graphs with the same vertex set but different edge sets, we construct a summary graph by adding the graphs together and deleting edges that occur fewer than three times. The dense subgraph $\{a,b,c,d\}$ appearing in the summary graph does not occur in any of the original graphs. (**b**) The vertices e and f are shared by cliques $\{a,b,c,d,e,f\}$ and $\{e,f,h,i\}$. The shared vertices can be assigned to both cliques only by approaches that are able to detect overlapping dense subgraphs (cliques are the densest subgraphs of a network)

functional modules [23, 51]. In fact, even recurrent paths are likely to correspond to functional modules [24]. In this section, we define and illustrate three types of recurrent patterns in unweighted graphs, our data mining algorithms to discover them, and their biological applications.

2.1 Coherent Dense Subgraphs

A straightforward approach to analyzing multiple networks is to aggregate these networks together and identify dense subgraphs in the aggregated graph. However, the aggregated graph can contain dense subgraphs that do not occur frequently, or even exist at all, in the original networks. Figure 6.1a illustrates such a case with a cartoon of six graphs. If we add these graphs together to construct a summary graph, we may find a dense subgraph containing vertices a, b, c, and d. Unfortunately, this subgraph is neither dense nor frequent in the original graphs. To overcome this problem, we propose looking for *Coherent Dense Subgraphs* that satisfy two criteria: (1) the nodes are densely interconnected, and (2) all of the edges should exhibit correlated occurrences in the whole graph set. In the following, we provide a formal definition of coherent dense subgraph and an algorithm to identify these patterns in multiple networks.

2.1.1 Problem Formulation

Consider a relation graph set D consisting of n undirected simple graphs: $D = \{G_i = (V,E_i)\}, i = 1,\ldots n, E_i \subseteq V \times V$. All graphs in the set share a common vertex set V. We denote the vertex set of a graph G by $V(G)$, and the edge set by $E(G)$. Let $w_i(u,v)$ be the weight of an edge $e_i(u,v)$ in G_i. For an unweighted graph, $w_i(u,v) = 1$ if there is an edge between u and v, otherwise $w_i(u,v) = 0$.

Definition 6.1 (Support). Given a relation graph set $D = \{G_1, G_2, \ldots, G_n\}$, where $G_i = (V, E_i)$, the support of a graph g is the number of graphs (in D) containing g as a subgraph. This measure is written $support(g)$. A graph is called *frequent* if its support is greater than a specified threshold.

Definition 6.2 (Summary Graph). Given a relation graph set $D = \{G_1, G_2, \ldots, G_n\}$, where $G_i = (V, E_i)$, the summary graph of D is an unweighted graph $\hat{G} = (V, \hat{E})$ containing only those edges present in at least k graphs of D. The parameter k is a user-defined support threshold (see an example in Fig. 6.1a).

Definition 6.3 (Edge Support Vector). Given a relation graph set $D=\{G_1, G_2, \ldots, G_n\}$, where $G_i = (V, E_i)$, the support vector $\mathbf{w}(e)$ of an edge e is of length n. The ith element of $\mathbf{w}(e)$ is the weight of edge e in the ith graph.

The support vector of edge (a,b) for the six graphs shown in Fig. 6.1a is $[1,1,1,0,0,0]$, while the support vector of edge (b,c) is $[0,0,0,1,1,1]$. Their support vectors clearly show that edges (a,b) and (b,c) are not correlated in this dataset, although both of them are frequent.

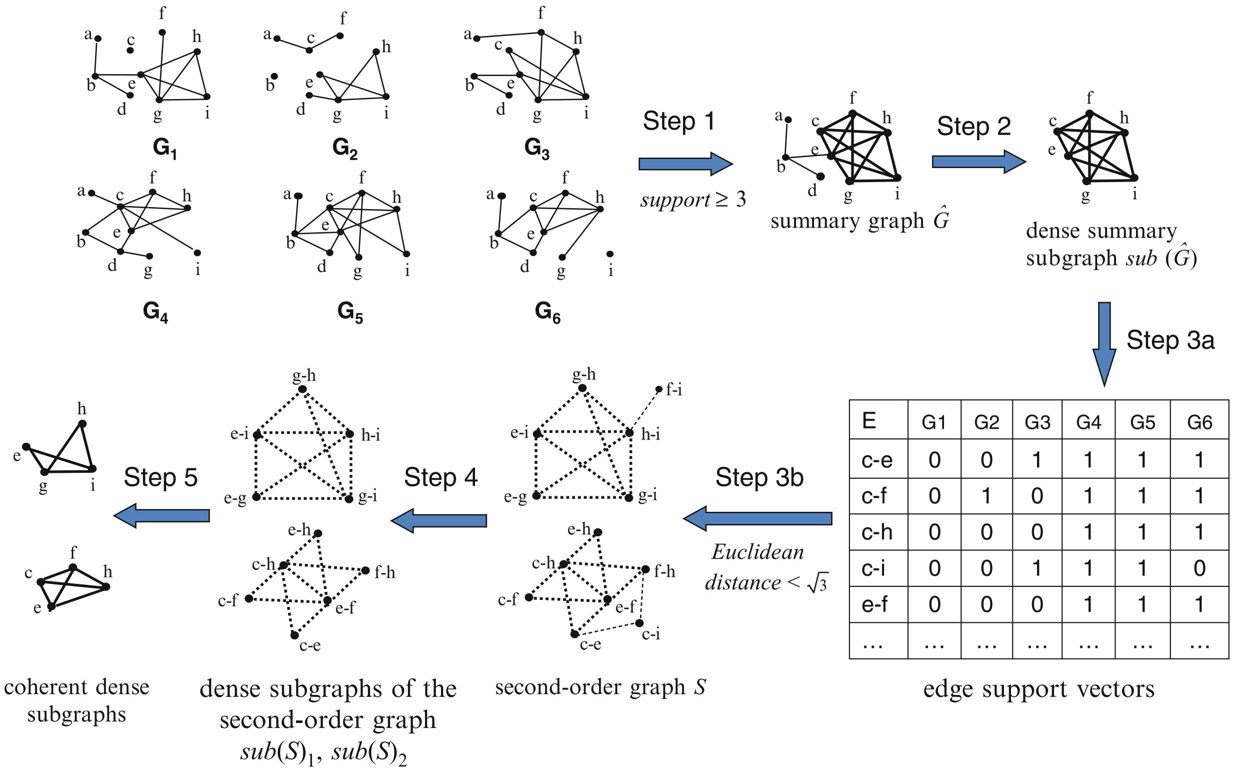
We use a special graph, the *second-order graph* S, to illustrate the co-occurrence of edges in a relation graph set D. Each edge in D is represented as a vertex in S. Two vertices u and v in S are connected if the edge support vectors $w(u)$ and $w(v)$ in D are sufficiently similar. Depending on whether or not the edges in D are weighted, the similarity measure could be the Euclidean distance or Pearson's correlation. Figure 6.2 (Step 3b) shows how to generate a second-order graph from a set of edge support vectors. For example, the Euclidean distance between the support vectors of edges (c,e) and (c,i) is only 1, so we create an edge between the vertices labeled (c,e) and (c,i) in the second-order graph S. This process is shown in Fig. 6.2. To contrast with the second-order graph, we term the original graphs G_i first-order graphs. This use of the second-order graph is just one type of second-order analysis, a concept proposed in one our previous publications [55].

Definition 6.4 (Second-Order Graph). Given a relation graph set $D=\{G_1, G_2, \ldots, G_n\}$, where $G_i = (V, E_i)$, the second-order graph is an unweighted graph $S = (V \times V, E_s)$ whose vertex set is equivalent to the edge set of G. In S, an edge is drawn between vertices u and v if the similarity between the corresponding edge support vectors $\mathbf{w}(u)$ and $\mathbf{w}(v)$ exceeds a specified threshold.

If the first-order graphs G_i are large and dense, S will be impractically large. To more efficiently analyze D, we construct second-order graphs S only for subgraphs of the summary graph $\hat{G}$.

Definition 6.5 (Coherent Graph). Given a relation graph set $D = \{G_1, G_2, \ldots, G_n\}$, where $G_i = (V, E_i)$, a subgraph $\mathrm{sub}(\hat{G})$ is coherent if all its edges have support greater than k and if the second-order graph of $\mathrm{sub}(\hat{G})$ is dense.

Definition 6.6 (Graph Density). The density of a graph g, written $density(g)$, is $\frac{2m}{n(n-1)}$, where m is the number of edges and n is the number of vertices in g.



E	G1	G2	G3	G4	G5	G6
c-e	0	0	1	1	1	1
c-f	0	1	0	1	1	1
c-h	0	0	0	1	1	1
c-i	0	0	1	1	1	0
e-f	0	0	0	1	1	1
…	…	…	…	…	…	…

Fig. 6.2 CODENSE: an algorithm to discover coherent dense subgraphs across multiple graphs (the dense subgraphs are marked with *bold* edges)

*The problem of mining **coherent dense subgraphs** can now be formulated as follows:* given a relation graph set $D = \{G_1, G_2, \ldots, G_n\}$, discover subgraphs g that satisfy the following two criteria: (1) g is a dense subgraph of the summary graph, and (2) g is coherent.

2.1.2 Algorithm

We have developed a scalable algorithm to mine coherent dense subgraphs [23]. It is based on the following two observations concerning the relationship between a coherent dense subgraph, the summary graph, and the second-order graph.

1. If a frequent subgraph of D is dense, then it must also exist as a dense subgraph in the summary graph. However, the converse is not true. A dense subgraph of the summary graph may be neither frequent nor dense in the original dataset (e.g., Fig. 6.1a).
2. If a subgraph is coherent (i.e., if its edges are strongly correlated in their occurrences across a graph set), then its second-order graph must be dense.

These two facts permit the mining of coherent dense subgraphs with reasonable computational cost. According to Observation 1, we can begin our search by finding all dense subgraphs of the summary graph. We can then single out coherent subgraphs by examining their corresponding second-order graphs. Our CODENSE algorithm consists of five steps, as outlined in Algorithm 1 and illustrated in Fig. 6.2. In Steps 2, 4 and 5, we employ a mining algorithm that allows for overlapping dense subgraphs (see Fig. 6.1b).

Step 1. CODENSE builds a summary graph by eliminating infrequent edges.

Step 2. CODENSE identifies dense subgraphs (which may overlap) in the summary graph. Although these dense subgraphs may not be frequently occurring in the original graph set, they are a superset of the true frequent dense subgraphs.

Step 3. CODENSE builds a second-order graph for each dense summary subgraph.

Step 4. CODENSE identifies dense subgraphs in each second-order graph. A high connectivity among vertices in a second-order graph indicates that the corresponding edges have high similarity in their occurrences across the original graphs.

Step 5. CODENSE discovers the real coherent dense subgraphs. Although a dense subgraph $\mathrm{sub}(S)$ found in Step 4 is guaranteed to have the co-occurrent edges in the relation graph set, those edges may not form a dense subgraph in the original summary graph. To eliminate such cases, we convert the vertices in $\mathrm{sub}(S)$ back to edges and apply the overlapping dense subgraph mining algorithm once more. The resulting subgraphs will satisfy both criteria for coherent dense subgraphs: (1) they are dense subgraphs in many of the original graphs, so all of their edges occur frequently; and (2) the support vectors of the edges are highly correlated across the relation graphs.

Algorithm 1: `CODENSE`

```
Step 1: build a summary graph $\hat{G}$ across multiple relation graphs $G_1, G_2, \ldots, G_n$;
Step 2: mine dense summary subgraphs $sub(\hat{G})$ in $\hat{G}$ using an overlapping dense subgraph mining algorithm;
foreach each dense summary subgraph $sub(\hat{G})$ do
    Step 3: construct the second-order graph $S$;
    Step 4: mine dense subgraphs $sub(S)$ in $S$ using an overlapping dense subgraph mining algorithm;
    Step 5: foreach each dense subgraph $sub(S)$ do
        convert $sub(S)$ into the first-order graph $G$;
        mine dense subgraphs $sub(G)$ in $G$ using an overlapping dense subgraph mining algorithm;
        output $sub(G)$;
    end
end
```

2.1.3 Experimental Study

We use coexpression networks derived from 39 yeast microarray datasets as a testing system for CODENSE. Each dataset comprises the expression profiles of 6,661 genes in at least eight experiments. These data were obtained from the Stanford Microarray Database [19] and the NCBI Gene Expression Omnibus [16]. The similarity between two gene expression profiles in a microarray data set is measured by Pearson's correlation. We transform Pearson's correlation (denoted r) into $\sqrt{\frac{(n-1)r^2}{1-r^2}}$, and model the latter quantity as a t-distribution with $n-2$ degrees of freedom (Here, n is the number of measurements used to compute Pearson's correlation). We then construct a relation network for each microarray dataset, connecting two genes if their Pearson's correlation is significant at the $\alpha = 0.01$ level. The summary graph $\hat{G}$ is then constructed by collecting edges with a support of at least six graphs. At all steps where dense subgraph mining is performed (see Algorithm 1), the density threshold is set to 0.4.

To assess the clustering quality, we calculated the percentage of functionally homogeneous clusters among all identified clusters. Based on the Gene Ontology (GO) biological process annotations, we consider a cluster to be functionally homogeneous if (1) the functional homogeneity modeled by the hypergeometric distribution [50] is significant at $\alpha = 0.01$; and (2) at least 40% of its member genes with known annotations belong to a specific GO functional category.

Within the hierarchical organization of GO biological process annotations, we define *specific functions* to be those associated with GO nodes that are more than five levels below the root. CODENSE identified 770 clusters with at least four

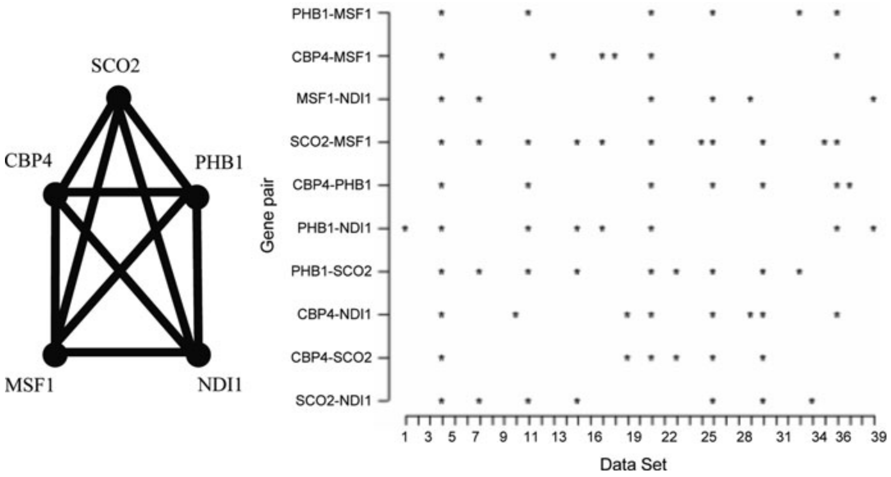


Fig. 6.3 The edge occurrence profiles of a five-gene clique in the summary graph

annotated genes. Of these clusters, 76% are functionally homogeneous. If we stop at Step 2 of the algorithm, obtaining dense subgraphs of the summary graph, only 42% are functionally homogenous. This major improvement in performance can be attributed to the power of second-order clustering as a tool for eliminating dense summary subgraphs whose edges do not show co-occurrence across the networks. As an example, consider the five-gene clique in the summary graph, {MSF1, PHB1, CBP4, NDI1, SCO2}, depicted in Fig. 6.3. The five genes are annotated with a variety of functional categories such as "protein biosynthesis," "replicative cell aging" and "mitochondrial electron transport," so the subgraph is not functionally homogenous. As it turns out, although all edges of this clique occur in at least six networks, their co-occurrence is not significant across the 39 networks (Fig. 6.3). Analyzing the second-order clusters can reveal such pseudoclusters, providing more reliable results.

The large set of functionally homogeneous clusters identified by CODENSE provides a solid foundation for the functional annotation of uncharacterized genes. Some of the clusters contain unknown genes, and if the dominating GO functional category is significantly overrepresented (Bonferroni-corrected hypergeometric p-value < 0.01), we can confidently annotate the unknown genes with that function. To assess the prediction accuracy of our method, we employed a "leave-one-out" approach: a known gene is treated as unknown before analyzing the coherent dense subgraphs, then annotated based on the remaining known genes in the cluster. We consider a prediction correct if the lowest common ancestor of the predicted and known functional categories is five levels below the root in the GO hierarchy. Note that the annotated yeast genes encompass 160 functional categories at level 6 of the GO hierarchy. We predicted the functions of 448 known genes by this method, and achieved an accuracy of 50%. With respect to truly unknown genes, we produced functional predictions for 169 genes, covering a wide-range of functional categories.

2.2 Frequent Dense Vertexset

Although CODENSE has been successfully applied to identify recurrent dense subgraphs across multiple coexpression networks, its criteria are too stringent to identify many potential recurrent coexpression clusters. CODENSE requires coherency of edge recurrence; that is, the entire edge set of a pattern has to show highly correlated recurrence across the graph set. However, edge occurrences in a coexpression network can be distorted by measurement noise or by the correlation threshold used to dichotomize the edges. In fact, any set of genes that is densely connected in a significant number of networks is likely to form a functional and transcriptional module, even if the edges differ from network to network. That is, as long as a consistently large percentage (e.g., $\geqslant 60\%$) of gene pairs in a gene set are connected in multiple networks, that gene set is considered as a recurrently dense pattern and is worthy of attention. We denote such patterns "frequent dense vertexsets" (FDVSs). In this section, we develop a method to efficiently and systematically identify FVDSs.

2.2.1 Problem Formulation

Given a graph $G = (V, E)$ and the subgraph induced by vertex set $V' \subseteq V$, written $G(V')$, we define the FDVS as follows,

Definition 6.7 (Frequent Dense Vertexset). Consider a relation graph set $D = \{G_1, G_2, \ldots, G_n\}$, where $G_i = (V, E_i)$ and each graph shares the vertex set V. Given a density threshold δ and a frequency threshold θ, $V' \subseteq V$ is a frequent dense vertexset if, among all induced graphs $\{G_i(V')\}$, at least $\theta|D|$ graphs have density $\geqslant \delta$.

According to the above definition, a FDVS is a set of vertices, rather than a classical graph with vertices and edges. This definition supports the concept of approximate graph patterns, which need not have exactly the same edge set in the supporting dataset. From a computational point of view, it could be hard to enumerate all of the frequent graphs that satisfy the density constraint. Therefore, we resort to an approximate solution that begins by aggregating the graphs into a summary graph and identifying its dense subgraphs in a top-down manner. The summary graph approach is straightforward, but suffers from two problems: (1) the edges in a dense summary subgraph may never occur together in the original graphs; and (2) noise in the graphs will also accumulate and may become indistinguishable from signals that occur only in a small subset of the graphs. We devised two techniques to overcome these problems. (1) Since similar biological conditions are likely to activate similar sets of transcription/functional modules, we enhance the signal of real patterns by partitioning the input graphs into groups of graphs sharing certain topological properties. Such groups are more likely to contain frequent dense vertexsets. Furthermore, by aggregating similar graphs the signal will be enhanced more than the noise. (2) For each group of graphs, we construct a *neighbor*

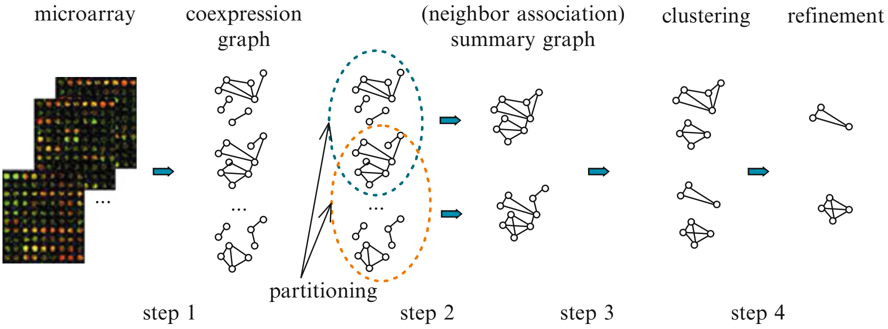


Fig. 6.4 The pipeline of our frequent dense vertexset mining algorithm (called NeMo). Step 1: extract coexpression graphs from multiple microarray datasets by removing insignificant edges. Step 2: partition the coexpression graphs into groups and construct a weighted summary graph for each group. Step 3: cluster each summary graph to identify dense subgraphs. Step 4: refine/extract frequent dense vertexsets from the dense subgraphs discovered in Step 3

association summary graph. This is a weighted graph, unlike the summary graph used by the CODENSE method. The edges of this graph measure the association between two vertices based on their connection strength with their neighbors across multiple graphs. For example, given two vertices u and v, if many small FDVSs include them, these two vertices are likely to belong to the same large FDVS. Figure 6.4 depicts the pipeline of this graph mining methodology. In the next subsection, we will examine this solution in detail.

2.2.2 Algorithm

Given n graphs, a frequent dense vertexset with density δ and frequency θ must form a subgraph with density $\geqslant \delta\theta n$ in the summary graph. According to this observation, we can begin by mining the dense subgraphs of the summary graph. The dense subgraphs are then processed to extract frequent dense vertexsets. This method is outlined as follows:

1. *Construct a summary graph:* Given n graphs, remove infrequent edges and then aggregate all graphs to form a summary graph S.
2. *Mine dense subgraphs from the summary graph:* Apply the overlapping dense subgraph mining algorithm to S. This step yields a set of dense subgraphs $\hat{M}$ satisfying some density constraint, for example, $\geqslant \delta\theta n$.
3. *Refine:* Extract true frequent dense vertexsets from each dense subgraph $\hat{M}$.

For the refinement step, we adopt a heuristic process. Given a dense summary subgraph $\hat{M}$ with n' vertices, we first calculate the weighted sum of the edges incident to each vertex. Next, we sort these n' vertices in ascending order of the weighted sum. Then, the vertices are removed from the list one by one until the remaining vertices in $\hat{M}$ form a frequent dense vertexset. This approach is referred to as a "greedy" refinement process. More advanced search methods such as simulated annealing can be applied as well. Starting from the dense subgraphs of a summary graph significantly reduces the search space, and provides a good starting point for the refinement process. On the other hand, it could generate false patterns. We offer two improvements to remedy this problem: (1) divide the original graphs into groups and formulate a series of summary graphs based on these groups of graphs. (2) Alter the weights in summary graph to reduce the impact of noisy edges.

To implement the first solution (partitioning the original graph set), we actually begin by mining each individual graph separately for dense subgraphs. The frequent subgraphs are then taken as seed vertexsets to bootstrap the mining process. This bootstrap process is as follows (see Fig. 6.4). (1) Extract dense subgraphs $\hat{M}$ from each individual graph. Then, refine these subgraphs for true frequent dense vertexsets M, using the greedy refinement process introduced above. (2) For each frequent dense vertexset M, calculate its supporting graph set $D_\delta(M) \subseteq D$. Take $D_\delta(M)$ as one subset. (3) Remove duplicate subsets. (4) For each unique subset $D_\delta(M)$, call the summary-graph-based approach introduced above to find frequent dense vertexsets in $D_\delta(M)$.

To implement the second approach (reweighting the summary graph), we introduce the concept of a *neighbor association summary graph*. Its intuition is as follows: given two vertices u and v in a graph, if many small frequent dense subgraphs contain both u and v, it is likely that u and v belong to the same cluster. In other words, if a graph/cluster is dense, then its vertices will share many dense subgraphs. Referring to the definition of a k-vertexlet provided below, let π_u be the set of frequent dense $(k-1)$-vertexlets that contain vertex u, and let $\pi_{u,v}$ be the set of frequent dense k-vertexlets that contain vertices u and v. We also define a scoring function $\text{score}(u,v)$ as follows,

$$\text{score}(u,v) = \frac{|\pi_{u,v}|}{\pi_u} \tag{6.1}$$

Definition 6.8 (Vertexlet). Given a vertex set V, a k-vertexlet is a subset of V with k vertices.

This scoring function is not symmetric: $\text{score}(v,u) \neq \text{score}(u,v)$. We take the average of the two scores, which is symmetric, as the weight of the edge between u and v. This new summary graph is called as the neighbor association graph because it relies on more than one neighbor to determine the weight between two vertices. This weighting method could increase the signal-to-noise ratio for identifying subtle dense subgraphs. The workflow for computing the neighbor association summary graph is outlined in Algorithm 1 of [51]. Once the neighbor association summary graph has been built, we apply the mining routine described above. The entire mining algorithm is named NeMo, for Network Module Mining.

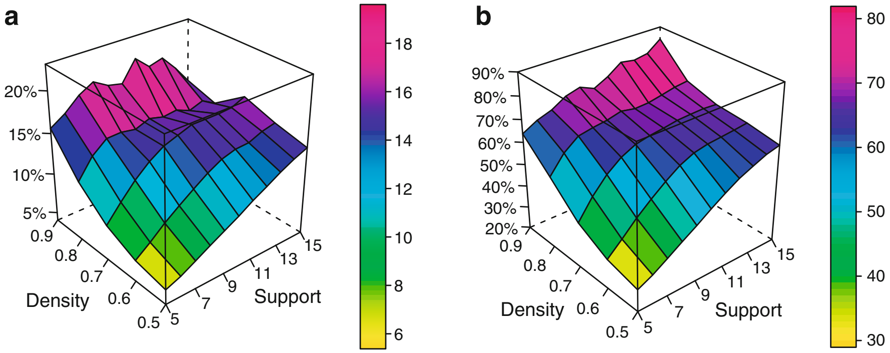


Fig. 6.5 Validation by ChIP-chip and GO data demonstrated that the likelihood of a coexpression cluster being a transcription module and functional homogeneous module increases significantly with its recurrence

2.2.3 Experimental Study

We selected 105 human microarray datasets, generated by the Affymetrix U133 and U95Av2 platforms. Each microarray dataset is modeled as a coexpression graph following the method introduced in Sect. 2.1.3. In this study, the most significant correlations with p-values less than 0.01 (the top 2%) are included in each graph. We applied NeMo to discover frequent dense vertexsets in these networks, and identified 4,727 recurrent coexpression clusters. Each cluster's density is greater than 0.7 in at least ten supporting datasets.

To assess the quality of the clusters identified by NeMo, we tested their member genes for enrichment of the same bound transcription factor. The transcription factors to target gene relationships were ascertained through ChIP-Chip experiments, which contain 9,176 target genes for 20 TFs covering the entire human genome. A recurrent cluster is considered a potential transcriptional module if (1) $>75\%$ of its genes are bound by the same transcription factor, and (2) the enrichment of the particular TF in the cluster is statistically significant with a hypergeometric p-value <0.01 relative to its genome-wide occurrences. Among the identified clusters, 15.4% satisfied both criteria. This is a high hit rate, considering we only tested for 1% of the approximately 2,000 transcription factors estimated to exist in the human genome. On average, the permuted set of clusters was enriched only 0.2% for a common transcription factor. This result demonstrates that our approach can reliably reconstruct regulatory modules. The integrity of the clusters is further validated by varying the threshold for density and recurrence. We find that as these criteria grow stricter, the proportion of identified clusters that share a common bound TF also increases (Fig. 6.5a).

The high quality of the clusters identified by NeMo is also supported by functional homogeneity analysis. We define a cluster to be functionally homogeneous if $>75\%$ of its member genes belong to the same Gene Ontology biological process with a hypergeometric p-value <0.01. As the cluster density and frequency thresholds increase, the functional homogeneity of the clusters increases as well (Fig. 6.5b). Among all identified clusters, 65.3% are functionally homogeneous compared to 2.2% of the permuted clusters.

2.3 General Recurrent Network Patterns

In the previous two sections, we focused on identifying recurrent dense subgraphs in multiple biological networks. Although such patterns often correspond to functional/transcriptional modules, there also exist many biological modules whose genes are not densely connected. Many types of relationships are possible among functionally-related genes – some lying beyond our current knowledge. These unknowns are exactly the reason why integrative analysis of multiple networks is such a powerful tool. Let us again use coexpression networks as examples. When we combine multiple expression networks, subtle signals may emerge that cannot be identified in any of the individual networks. Such signals include recurrent paths that may extend beyond simple coexpression clusters yet represent functional modules. If we only consider a single coexpression network, it is difficult to stratify functionally important paths from their complex network environment. However, if a path frequently occurs across multiple coexpression networks, it is easily differentiated from the background. In this section, we describe our method to systematically identify recurrent patterns of any kind from multiple relation graphs.

2.3.1 Recurrent Network Pattern Discovery Algorithm

To identify frequently occurring network patterns, we design a data mining procedure based on frequent itemset mining (FIM) and biclustering methods. Given n relation graphs, we wish to identify patterns that comprise at least four interconnected nodes and occur in at least five graphs. This is computationally very difficult due to the large number of potential patterns. Our approach first searches for frequent edge sets that are not necessarily connected, then extracts their connected components. Conceptually, we formulate the n graphs as a matrix where each row represents an edge (i.e., a gene pair), each column represents a graph, and each entry (1 or 0) indicates whether the edge appears in that graph. In this framework, the problem of discovering frequent edge sets can be formulated as a biclustering problem that searches for submatrices with a high density of 1's. This is a well-known NP-hard problem.

We have developed a biclustering algorithm based on simulated annealing to discover frequent edge sets. We employ simulated annealing to maximize the objective function $\frac{c'}{mn+\lambda c}$, where c is the number of 1's in the input matrix, c_0, m and n are the numbers of 1's, rows and columns in the bicluster, respectively, and λ is a regularization factor. Clearly, this objective function favors large biclusters with a high density of 1's. Note that the density is maximized (to unity) when $c' = mn$, while the size of bicluster is maximized when $c' = c$ (i.e., the pattern is as large as the input matrix). The regularization parameter λ controls the trade-off between density and size. However, there is no theoretical result on suggesting an optimal value for λ. In this study, we tried many heuristic choices of λ. The reported results are based on $\lambda = \frac{0.2}{\max(1,\log_{10}(n_1))}$, where n_1 is the number of edges in the initial configuration (i.e., the seed).

Although this method performs well in our experiments, the enormous search space (the edge/graph matrix has more than 1 million rows and 65 columns) has to be restricted to discover hundreds of thousands of patterns in a reasonable time frame. To address this problem and generate seeds for our biclustering algorithm, we employ the FIM technique [20]. Below, we briefly describe FIM and related concepts.

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of items and let D be a database of m transactions. Each transaction T is a set of items such that $T \subseteq I$. Supposing X is a set of items $X \subseteq I$, a transaction T is said to contain X if and only if $X \subseteq T$. The itemset X is called frequent if at least s transactions in the database contain X. The output of a standard FIM algorithm is a list of all possible itemsets X which occur in at least s transactions. In our case, we can regard an edge as a transaction and its occurrence in a particular graph as an item. Given our frequency constraint, we need only include edges occurring in at least five graphs in the transaction dataset. Note that the frequent itemsets and their supporting transactions are actually submatrices (biclusters) full of 1's. These clusters with perfect density can serve as seeds for our biclustering algorithm, which searches for larger biclusters that permit holes (i.e., 0's). The FIM algorithm may produce millions of itemsets which contain at least four edges and occur in at least five graphs. These patterns should not be used directly as seeds, however, because they overlap a great deal. This problem is well-known in the data mining community. To improve the seed patterns and reduce unnecessary computation in the biclustering algorithm, we first remove all FIM patterns whose supporting transactions/edges are a subset of some other pattern. Second, we merge two patterns if the resulting submatrix has a density larger than 0.8. This procedure is repeated until no additional merger can happen.

After this postprocessing, we will have about half a million patterns to feed our biclustering algorithm. Given a FIM pattern with v genes, we generate a matrix of all possible edges ($\frac{v(v-1)}{2}$) and all datasets. This matrix serves as a seed for the biclustering algorithm, and is also used as the initial configuration in the algorithm's simulated annealing procedure. Finally, we extract connected components from each output bicluster produced by our algorithm.

2.3.2 Predicting Gene Functions from Recurrent Network Patterns

Given a network pattern, the most popular schemes for predicting gene function employ the hypergeometric distribution to model the probability of genes function based on neighborhood. However, this method ignores the network topology of the recurrent patterns, which is probably their most important aspect. To avoid this problem, we have developed a new method of estimating gene function based on random walks through the graph that can fully explore the topology of network patterns. Our method is still based on the principle of "guilt by association." In terms of network topology, the degree of association between two genes can be measured by how close they are (i.e., the length of the path between them) and how tightly connected they are (i.e., the number of paths existing between them). Statistically, they translate into the likelihood of reaching one gene from another gene in a random walk. This probability can be approximated by matrix multiplication.

Given a network pattern consisting of v genes, let P be a stochastic matrix of size $v \times v$. The element P_{ij} is $1/n_i$ if genes i and j are connected and 0 otherwise, where n_i is the number of neighbors of gene i. If we regard the genes as states and P_{ij} as the probability of gene/state i transforming into j, then a random walk through the graph can be thought of as a Markov process. From this perspective, it is easy to see that each element of the matrix P^k is the probability that gene i reaches gene j in a k-step random walk. The intuition behind our method is that genes with similar functions are more likely to be well connected (i.e., gene i will reach gene j with high probability in a random walk). Simply put, we expect the probability P_{ij}^k to be large if genes i and j share the same function. Let o be the Gene Ontology binary matrix, where element o_{ij} is 1 if gene i belongs to category j and 0 otherwise. Then, the matrix $M = P^k o$ gives the *network topology scores* of genes relating to functional categories. The higher this score, the more likely a gene has that function. In practice, we choose $k = 3$ to confine our prediction to a local area of network patterns. The function of each gene is estimated as the functional category with the maximum score in the corresponding row of the score matrix M.

In an attempt to improve our method, we tried including attributes other than the network topology scores of a network pattern in the final prediction. These attributes are, recurrence, density, size, average node degree, the percentage of unknown genes, and the functional enrichment of network modules. We use a random forest[1] to determine whether function assignments based on the network topology score are robust. In other words, the purpose of the random forest is to determine whether to accept or reject a functional assignment based on the network topology score. The random forest was trained using the assignments of known genes. The trained model was then applied to classify unknown genes. We only keep the function assignments that the random forest classified as "accept."

2.3.3 Experimental Study

We collected 65 human microarray datasets, including 52 Affymetrix (U133 and U95 platforms) datasets and 13 cDNA datasets from the NCBI Gene Expression Omnibus [16] and SMD [19] databases (December 2005 versions). Each microarray

[1] A random forest is a collection of tree-structured classifiers [8].

dataset is modeled as a coexpression graph following the procedure introduced in Sect. 2.1.3. The FIM and biclustering algorithms described above yield a total of 1,823,518 network patterns (modules) which occur in at least five graphs. After merging patterns with similar network topologies and dataset recurrence, we are left with 143,400 distinctive patterns involving 2,769 known and 1,054 unknown genes. The sizes of the patterns vary from 4 to 180.

We define a module to be functionally homogenous if the hypergeometric p-value after Bonferroni correction is <0.01. Among the identified network patterns, 77.0% are functionally homogenous by this standard. In general, patterns that occur more frequently are more likely to be functionally homogenous. This observation supports our basic motivation for using multiple microarray datasets to enhance functional inferences, namely that by considering pattern recurrence across many networks we can enhance the signal of meaningful structures. We identify network modules with a wide-range of topologies. In fact, 24% of the modules have connectivities <0.5.

To explore the relationships among the network members other than coexpression, we resort to the only available large-scale source: protein interaction data. We retrieved human protein interaction information from the European Bioinformatics Institute (EBI)/IntAct database [21] (version 2006-10-13). For each of the 143,400 detected patterns, we then tested whether protein interactions were overrepresented in member genes compared to all human genes using the hypergeometric test to evaluate significance. A total of 60,556 (22.44%) patterns were enriched in protein interaction at a p-value of 0.001 level. This shows that genes belonging to a module are much more likely to encode interacting proteins. Interestingly, many of the protein-interaction-enriched network modules also fall into functional categories such as protein biosynthesis, DNA metabolism, and so on. There are even many cases where the interacting protein pairs are not coexpressed.

For each of the 143,400 recurrent network patterns, we identified the function of each member gene with the maximum network topology score. We then trained a random forest and made functional predictions for 779 known and 116 unknown genes with 70.5% accuracy. It should be noted that the potential prediction accuracy of this method is probably much higher; the rate of 70% is due to the sparse nature of human GO annotations. Since GO annotations are based only on positive biological evidence, it is likely that many annotated genes have undiscovered functions. Furthermore, the GO directed acyclic graph structure is not perfect.

Since our approach allows a given gene to appear in more than one network module, we are able to perform context-sensitive functional annotation. That is, we can associate each gene multiple functions as well as the network environments in which the gene exerts those functions. These contexts and relationships represent valuable information, even if all of a gene's function are already known. Among our predictions, 20% of genes are assigned multiple functions. This rate is almost certainly an underestimate, since for each network module, we only annotated genes with a single functional category: the one associated with the highest network topology score.

3 Differential Network Patterns

Suppose that a set of biological networks is divided into two classes, for example, those related to a specific disease and those obtained under normal or unrelated conditions. It is then interesting to identify network patterns that differ significantly between these two classes. In fact, it has become clear that many complex conditions such as cancer, autoimmune disease, and heart disease are characterized by specific gene network patterns. Recently, we designed an integrative approach to inferring network modules specific to a phenotype [33]. A series of microarray datasets modeled as coexpression networks is labeled with phenotypic information such as the type of biological sample, a disease state, a drug treatment, etc. For each phenotype, we can partition all microarray datasets into a positive class of datasets appropriately annotated with that phenotype, and a background class containing the rest of the datasets. We have designed a graph-based simulated annealing approach [26] to efficiently identify groups of genes that form dense subnetworks preferentially and repeatedly in a phenotype's positive class. Using 136 microarray datasets, we discovered approximately 120,000 modules specific to 42 phenotypes and developed validation tests combining Gene Ontology, Gene Reference Into Function (GeneRIF) and UMLS data. Our method is applicable to any kind of abundant network data with well-defined phenotype associations, and paves the way for a genome-wide atlas of gene network–phenotype relationships.

3.1 Problem Formulation

Consider a relation graph set $D = \{G_1, G_2, \ldots, G_n\}$, where each graph $G_i = (V, E_i)$ is annotated with a set of phenotypes. For each phenotype, we partition D into a positive class D_{P} consisting of graphs annotated with that phenotype and a background class $D_{\mathrm{P}}^{c} = D \setminus D_{\mathrm{P}}$. Our problem is to identify groups of genes which form dense subgraphs repeatedly in the phenotype positive class but not in the background class. More specifically, we aim to satisfy three criteria: first, a gene set must be densely connected in multiple graphs; second, the annotations of these graphs must be enriched in a specific phenotype; and third, the gene set meeting the first two criteria must be as large as possible. Put simply, this problem is to find modules with three qualities: density, phenotype specificity, and size.

For the first criterion, we consider a gene set to be densely connected if its density is larger than a hard threshold (typically 0.66). However, because we will use simulated annealing as the optimization method (see Sect. 3.2), hard thresholds are too restrictive. Rather, we want the algorithm to accept intermediate states that may be unfavorable. We, therefore, design an objective function f_{dens} with a soft threshold, where unfavorable values of the density increase the cost exponentially. This objective function is defined in (6.3) below. Similarly, the other two criteria also use soft thresholds in their objectives. The second criterion (specificity) states that given a phenotype, we wish to find dense gene sets that occur frequently in the positive class but infrequently in the background class. The specificity objective function is defined in (6.4). It uses the hypergeometric test to quantify the significance of phenotype enrichment and favors low p-values, again at an exponential rate. For simplicity and computational considerations, we limited the size of the module to 30 genes. We believe this to be an ample margin for phenotypically relevant gene sets. Equation (6.2) shows the size objective function, which contains both a linear component (first term) and an exponential component (second term). The exponential component sets a strong preference for low sizes (four to five vertices), but the linear component continues to reward size increases above this soft threshold.

We supplemented the three main objectives with a fourth: the *density differential* defined in (6.5). This term compliments the density and specificity objective functions by comparing the average density of the cluster in the background datasets to its density in the phenotype datasets. The rationale behind this term is as follows. Since the specificity objective function only takes a state's active datasets as arguments, the transition to a neighboring state may yield a sudden change in the specificity energy because its active datasets are different. However, many neighboring states can have subtle changes in the density distribution among the active and inactive datasets that is not captured by the density and specificity functions alone. The density differential function is, therefore, designed to reward these subtle density changes, helping direct the simulated annealing process toward more phenotype-specific clusters. We found that using the density differential in combination with the specificity and density allowed the algorithm to converge faster and find better clusters than either option alone.

The individual objective functions that we designed take the following forms:

$$f_{\text{size}}(x) = \exp\left\{-\alpha\left(\frac{|x|}{\gamma} - o_s\right)\right\} \tag{6.2}$$

$$f_{\text{dens}}(x) = \exp\left\{-\alpha\left(\min_{i \in D_{\text{A}}}\left(\delta_i(x)\right) - o_\delta\right)\right\} \tag{6.3}$$

$$f_{\text{spec}}(x) = \log\left(\mathbb{P}\left(Y \geq |D_{\text{A}} \cap D_{\text{P}}|\right)\right) \tag{6.4}$$

$$f_{\text{diff}}(x) = \left(\frac{1}{|D_{\text{P}}^c|}\sum_{i \in D_{\text{P}}^c}\delta_i(x) - \frac{1}{|D_{\text{P}}|}\sum_{i \in D_{\text{P}}}\delta_i(x)\right) \tag{6.5}$$

where

D_{P} is the set of graphs annotated with the current phenotype,
D_{A} is the set of graphs in which the gene cluster is dense,
and $Y \sim$ hypergeometric$\left(|D_{\text{A}}|, |D_{\text{P}}|, |D_{\text{P}}^c|\right)$.

The exponential components of these functions prevent the simulated annealing algorithm from settling on an extreme case with just one of the desired qualities (such as a very specific triangle, which is always very dense and small). Improvements to such cases are always rewarded, however, and they are accepted as intermediate steps with good probability. We selected the parameters $\alpha = 20$, $\gamma = 30$, $o_\delta = 0.85$, and $o_s = 0.2$ based on our simulation results comparing biologically validated clusters with clusters arising from random chance.

We combined the four objective functions into a single function using a weighted sum $f(x) = w_1 f_{\text{size}}(x) + w_2 f_{\text{dens}}(x) + w_3 f_{\text{spec}}(x) + w_4 f_{\text{diff}}$. The key difficulty with this approach is determining an appropriate set of weights. In previous studies, this has been accomplished empirically [13]. We do the same, for the following reasons. First, we are interested in finding a single optimal or near-optimal objective function, rather than exploring the extremes of each term. Second, the overall effectiveness of our algorithm turns out to be consistent for a wide-range of weights. Finally, although we chose weights based on the algorithm's performance with simulated data, it also behaved well on real data. The weights for size, density and specificity, and density differential are 0.05, 0.05, 5, and 50, respectively.

3.2 Differential Network Pattern Discovery Algorithm

As stated above, we use simulated annealing (SA) to identify differential patterns. This well-established stochastic algorithm has been successfully applied many other NP-complete problems [44]. Our specific design for the SA algorithm follows.

3.2.1 Search Space

A *state* is defined as a set of vertices, and the search space is the set of all possible states. For simplicity and computational considerations, we limit the space to sets with fewer than 30 vertices. We believe this to be an ample margin for phenotypically relevant gene sets. Formally, we define the search space as $\mathscr{S} = \{x : x \subset V, |x| \leq 30, |x| \geq 3\}$.

3.2.2 Differential Coexpression Graphs

To dramatically increase the probability of finding optimal modules across many massive networks, we wish to narrow down the search space. We, therefore, construct a weighted *differential coexpression graph* for each phenotype. This graph summarizes the differences between gene coexpression networks in the phenotype class and those in the background class. The differential coexpression graph is used by the SA algorithm to create neighboring states (see Sect. 3.2.4).

The weighted differential coexpression graph $G_\Delta = (V, E_\Delta)$ contains only edges (coexpression relationships) that are present frequently in D_{P} but infrequently in D_{P}^{c}. The specificity of a single edge can be measured by the significance p of a hypergeometric test comparing the abundance of the edge in D_{P} to its overall abundance in D. The vertex set V of G_{Δ} is the same as that of D, and the weight of an edge is $-\log(p)$. In this way, heavier edges in this graph represent pairs of genes that exhibit elevated coexpression highly specific to D_{P}.

3.2.3 Initial States

SA attempts to find a global optimum state. If we were to use random initial states and run the algorithm for a long time, we will always arrive at approximately the same final state: the largest vertex set having the most evidence for coexpression and phenotype specificity. However, we are interested in finding many independent vertex sets. We, therefore, designed a systematic way of generating initial states ("seeds") and restricted the SA search space to vertex sets containing these seeds.

We define a *triangle* as a set of three vertices that is fully connected in at least one dataset. The hypothesis underlying our strategy is that if a set of genes is coexpressed specifically in datasets annotated with the phenotype of interest, then this set will include at least one triangle that appears frequently in the positive class and rarely in the background class.

Therefore, for each phenotype we tested every triangle appearing in the positive class for enrichment (using the hypergeometric test) with respect to the background class. For each triangle with a hypergeometric p-value less than 0.01, we ran the SA algorithm with the constraint that states must be supersets of the initial triangle.

3.2.4 Selection of Neighboring States

We define a *neighbor* of the current state as any state containing either one more or one fewer vertex. We create neighboring states by first determining whether to add or remove a vertex, then choosing the vertex based on an appropriate probability distribution.

If a cluster has size 3, it consists only of the initial seed so a vertex must be added. If a cluster has size 30 (maximum), a vertex must be removed. For intermediate values, we proceed as follows.

Let x be the current state. We narrow the choice of vertices to be added by considering only those with at least one edge to a vertex in x in at least one of the phenotype datasets. This criterion is easily justified, as no other vertices could possibly contribute to x as a dense, phenotype-specific cluster, even as an intermediate step. It can be shown that this set corresponds exactly to $\mathcal{N}_x = \{g : g \notin x, \sum_{h \in x} w_{\Delta}(g,h) > 0\}$ (See Sect. 3.2.2).

The probability of removing a vertex is given by $p_{\mathrm{rem}} = s_0/|\mathcal{N}_x|$, where s_0 is an estimate of how many vertices will improve the state. This simple function allows the SA process ample time to consider many neighbors before attempting to remove a vertex, since the number of neighboring vertices vastly outnumbers the number of vertices in a cluster. We heuristically chose $s_0 = 20$ as an appropriate average number. In the future, an iterative estimation of s_0 as the average size of the returned clusters might improve the performance of the algorithm.

In the event that a gene is to be removed, it is chosen uniformly from the cluster. When adding a gene, however, the probability of selecting vertex $g \in \mathcal{N}_x$ is proportional to the summed weights of edges in the differential coexpression graph leading from g to members of x. Formally, we have: $\mathbb{P}(g_a \text{ is added}) = \sum_{a \in x} w_\Delta(g_a, a) / \sum_{b \in \mathcal{N}_x} \sum_{a \in x} w_\Delta(a, b)$.

3.2.5 Annealing Schedule

We used the schedule $T_k = T_{\max} / \log(k+1)$, where k is the iteration number and T_k is the temperature at that iteration [18]. The initial temperature for our study was 4. This schedule form guarantees optimality for long run times. Although it might be argued that long run times are impractical, we found that for an identical number of iterations, this schedule resulted in lower-energy clusters than the oft-used exponential schedule $T_{k+1} = \alpha T_k = \alpha^k T_{\max}$. We ran the algorithm for a maximum of 1,000,000 iterations or until the simulated annealing converged. In cases where the maximum number of iterations was reached, we forced convergence to the best local minimum by a near-greedy exploration of the neighborhood, achieved by decreasing the temperature to near zero.

3.2.6 Postfiltering

Recall that we forced the initial seed triangle to be part of the final result. Clearly, some of these seeds will result from noise alone, in which case the final output will not be biologically significant. To remove these clusters, we discarded any vertex set not meeting the following criteria: size greater than 6, density greater than 0.66, and FDR-corrected phenotype specificity (p-value) less than 0.01. Moreover, the cluster must be dense in at least three datasets related to the target phenotype. After filtering, we merged redundant clusters with intersections/unions greater than 0.8.

3.3 Experimental Study

We selected microarray datasets from NCBI's Gene Expression Omnibus [16] that met the following criteria: all samples were of human origin, the dataset had at least eight samples (a minimum for accurate correlation estimation), and the platform was either GPL91 (Affymetrix HG-U95A) or GPL96 (Affymetrix HG-U133A). Throughout this study, we only considered the 8,635 genes shared by both platforms (and therefore all datasets). All 136 datasets meeting these criteria on 28 Feb 2007 were used for the analysis described herein.

We determined the phenotypic context of a microarray dataset by mapping the Medical Subject Headings (MeSH) of its PubMed record to UMLS concepts. This process is more refined than scanning the abstract or full text of the paper, and in practice results in much cleaner and more reliable annotations [9, 10]. UMLS is the largest available compendium of biomedical vocabulary, with definitions and hierarchical relationships spanning approximately one million interrelated concepts. The UMLS concepts include diseases, treatments, and phenotypes at various levels of resolution (molecules, cells, tissues, and whole organisms). To infer higher-order links between datasets, we annotated each dataset with all matching UMLS concepts and their ancestor concepts. The datasets received a total of 467 annotations, of which 80 mapped to more than five datasets. Some of the latter were mapped to identical sets of datasets; after merging these, we were left with 60.

For each dataset, we used the Jackknife Pearson correlation as a measure of similarity between two genes (the minimum of the leave-one-out Pearson correlations). To create the coexpression network, we selected a cutoff corresponding to the 150,000 strongest correlations (0.4% of the total number of gene pairs: $\binom{8,635}{2} \approx 3.73 \times 10^7$). This choice was motivated by exploring the statistical distribution of pairwise correlations, which we do not detail here.

We applied our simulated annealing approach to all 136 microarray datasets covering 42 phenotype classes. The phenotypes related to a wide-range of diseases (e.g., leukemia, myopathy, and nervous system disorders) and tissues (e.g., brain, lung, and muscle). The procedure described above identified 118,772 clusters that satisfied our criteria for a concept-specific coexpression cluster. The number of clusters found for a given phenotype increased with the number of datasets annotated with that phenotype: most of the phenotypes with only a few associated datasets yielded few clusters. The most strongly represented phenotype was “nervous system disorders,” with 15 associated datasets and 22,388 clusters.

We used two different methods to evaluate cluster quality. First, we assessed the functional homogeneity of a cluster by testing for enrichment for specific Gene Ontology [14] biological process terms. If a cluster is enriched in a GO term with a hypergeometric p-value less than 0.01, we consider it functionally homogeneous. Of the 118,772 clusters derived from all phenotypes, 78.98% were functionally homogenous. This validation demonstrates a key advantage of our approach: by focusing on clusters specific to a phenotypically related subset of all datasets, we are less likely to detect constitutively expressed clusters such as those consisting of ribosomal genes or genes involved in protein synthesis.

While the GO database provides information on a gene’s functions, it fails to describe its phenotypic implications. To map individual genes to phenotypes, we used GeneRIF [34]. This database contains short statements derived directly from publications describing the functions, processes, and diseases in which a gene is implicated. We mapped the GeneRIF notes to UMLS metathesaurus terms (as with the dataset MeSH headings), then annotated genes with the UMLS concepts. Similar

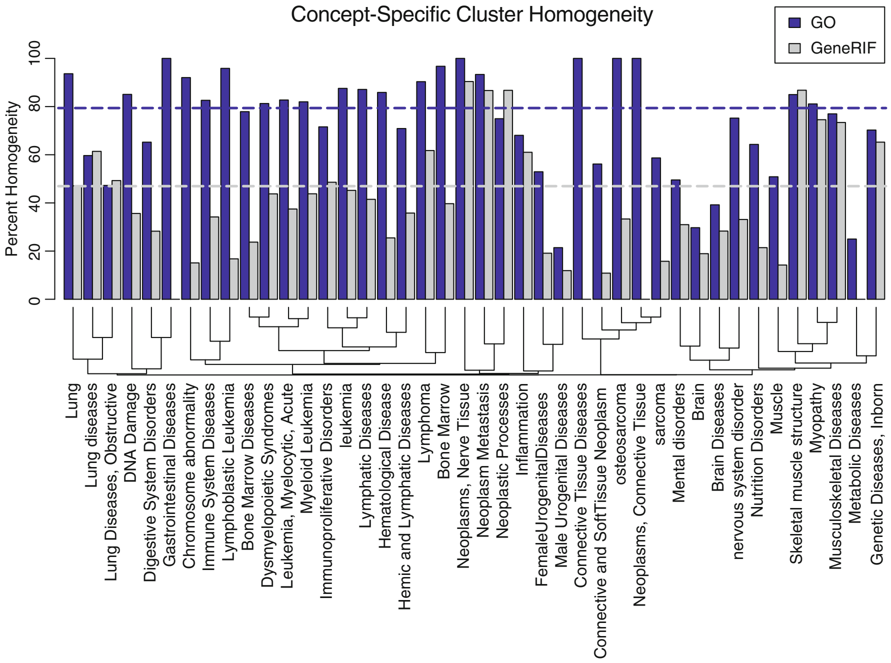


Fig. 6.6 Cluster homogeneity by phenotype. For each phenotype, the proportion of clusters that are significantly enriched (p-value <0.01) for a GO biological process (*blue*) or a GeneRIF UMLS concept (*gray*). The *dotted lines* show the overall homogeneity for all clusters. The dendrogram shows the distance between phenotypes in terms of dataset overlap

to our analysis of the GO annotations, we then assessed the *conceptual homogeneity* of gene clusters in specific UMLS keywords with the hypergeometric test, enforcing a p-value of 0.01 or less. The proportion of conceptually homogeneous modules was 48.3%. Clusters are less likely to have conceptual homogeneity than functional homogeneity, probably due to a dearth of GeneRIF annotations. In some situations, however, GeneRIF performs better. For example, many cancer-related phenotypes such as "Carcinoma," "Neoplasm Metastasis," and "Neoplastic Processes" are more likely to have GeneRIF homogeneity. This effect could be attributed to the abundance of related literature. The functional and conceptual homogeneity of clusters derived from different phenotype classes is summarized in Fig. 6.6.

In addition to testing for functional and conceptual homogeneity, we assessed whether the clusters were involved in the phenotype condition in which they were found. Again, we used both GO and GeneRIF independently for this.

Recall that each functionally homogeneous module is associated with one or more GO biological functions, and also with the phenotype in which it was found. We summarize the GO functions by mapping them to "informative nodes," a concept we introduced in our earlier work [54]. We then tested them for overrepresentation in that phenotype class. This provided, for each of 33 phenotypes (out of 42 phenotypes having at least one module), a list of gene module functions that are active in that phenotype more often than expected by chance. Many of these GO functions are clearly related to the phenotype in which they were found. For example, the phenotype “Mental disorders” has three GO biological processes related to brain function: “synaptic transmission” (2.3e–62), “neuron differentiation” (5.4e–42), and “central nervous system development” (7.9e–25). Our approach also identifies biological processes related to tissue phenotypes. For example, the “Skeletal muscle structure” phenotype is significantly enriched with modules that are homogeneous in the biological functions “muscle system process” (4.0e–221), “actin filament-based process” (1.23e–150), and “skeletal development (1.53e–03).” The functional association between a module’s GO function and the phenotype in which it is active suggests that our clusters are indeed linked to the phenotype conditions in which they were identified. In addition to GO informative nodes, we also tested each phenotype for overrepresentation of UMLS concepts from GeneRIF. This overrepresentation shows which diseases, tissues, and biological concepts are significantly enriched in each phenotype. In Table 6.1, we highlight some of these overrepresented functions and concepts.

The preceding analysis relies on our subjective evaluation of matches between UMLS and GO terms. We can conduct a more objective analysis using the GeneRIF data, which can be mapped directly to the same UMLS terms used to classify phenotypes. We counted the modules that were conceptually homogeneous with respect to the UMLS annotations that defined their respective phenotype classes. Of the 42 phenotypes represented in our study, 26 had one or more matching modules. The proportion of matching modules among total modules in these 26 phenotypes ranged from 0.04 to 33.6%. Although these numbers may not sound impressive, these proportions are significantly larger than expected by chance. We used a permutation test to assess the statistical significance of our analysis. We randomly assigned existing clusters to one of the 47 phenotypes with at least one cluster, while holding the number of clusters assigned to each phenotype constant. One million of these permutations were generated. Thirteen of the phenotypes were found to be significantly enriched with conceptually homogenous modules after FDR correction. They are shown in Table 6.2. The high significance for many of the phenotypes indicates that the low percentages are probably due to a dearth of GeneRIF annotations. As GeneRIF becomes more comprehensive, we expect the performance to improve in both the percentage of matching clusters and the number of phenotypes that are significant. We also found that the UMLS text mining of the GeneRIF database and the MeSH headers is not perfect, so improvements and refinements in those areas should also improve our validation results.

Table 6.1 Selected UMLS Concepts and their principal annotations. We annotated clusters using Gene ontology and GeneRIF, as detailed in the text. We then identified the annotations that were preferentially found in one concept relative to the others, as assessed by the hypergeometric test (Bonferroni-corrected *p*-values shown in parentheses)

Concept	Total	Over-represented GO annotations	Over-represented GeneRIF annotations
Lymphoma	890	Cell cycle phase (9.2e–276)	Lymphoreticular tumor (2.6e–93)
		Cell cycle checkpoint (1.2e–14)	Abnormal Hematopoietic and lymphoid cell (2.6e–22)
		Regulation of cell cycle process (3.2e–08)	Low grade B-cell lymphoma morphology (3.5e–19)
		Antigen processing and presentation (7.7e–03)	
Mental disorders	866	Synaptic transmission (2.3e–62)	Schizophrenia (4.3e–12)
		Neuron differentiation (5.4e–42)	Neurons (1.2e–11)
		Central nervous system development (7.9e–25)	Alzheimer’s disease (3.4e–04)
Muscle	584	Muscle system process (7.9e–52)	Heart (1.2e–20)
			Intrathoracic cardiovascular structure (3.1e–19)
			Muscle, striated (8.2e–15)
Myopathy	6,328	Actin filament-based process (7.2e–21)	Coronary heart disease (<1e–324)
		Muscle system process (4.6e–06)	Disorder of skeletal muscle (<1e–324)
Neoplastic processes	1,486	Keratinocyte differentiation (<1e–324)	Lung neoplasms (2.6e–207)
		Cell cycle checkpoint (1.0e–124)	Triploidy and polyploidy (2.8e–179)
		Regulation of mitotic cell cycle (7.4e–122)	Tumor of dermis (8.2e–123)
		Cell division (1.7e–107)	Glioma (6.4e–121)
Skeletal muscle structure	6,719	Muscle system process (4.0e–221)	Musculoskeletal structure of limb (4.3e–46)
		Actin filament-based process (1.2e–150)	Heart (7.6e–46)
		Skeletal development (1.5e–03)	

Table 6.2 Phenotypes for which the annotated clusters are consistent with the phenotype class in which they were derived. The first column indicates a UMLS phenotype. The second column displays the total number of clusters active in that phenotype class. The third and fourth columns show the percentage of clusters annotated with that phenotype in the phenotype class and in the background class, respectively. The fifth column shows the FDR-corrected *p*-value for the difference between the classes. The statistical significance was calculated by permuting the clusters across the dataset phenotypes 1,000,000 times. Concepts with a *p*-value less than 4.7e–6 were never outperformed by the permutations

Phenotype	Total clusters in phenotype class	Matching clusters in phenotype class (%)	Matching clusters in background class (%)	*p*-value
Mental disorders	791	3.12	0.17	<4.7e–06
Lymphoma	409	20.11	0.97	<4.7e–06
Myopathy	645	15.46	3.65	<4.7e–06
Musculoskeletal diseases	1,619	2.26	1.33	<4.7e–06
Genetic diseases, inborn	1,470	7.86	1.82	<4.7e–06
Neoplasms, nerve tissue	765	33.60	2.02	<4.7e–06
Neoplastic processes	794	9.08	4.19	<4.7e–06
Nervous system disorder	2,214	4.44	2.69	<4.7e–06
Skeletal muscle structure	154	0.94	0.18	<4.7e–06
Hemic and lymphatic diseases	1,129	1.17	0.65	1.3e–05
Bone marrow diseases	523	1.31	0.52	5.3e–03
Leukemia	460	0.55	0.36	2.9e–02
Muscle	483	1.03	0.31	3.5e–02

4 A Computational Model for Multiple Weighted Networks: Tensor

In previous sections, we approached the analysis of multiple large networks through a series of heuristic, graph-based, data mining algorithms. While useful, this class of methods faces two major limitations. (1) The general strategy is a stepwise reduction of the large search space, but each step involves one or more arbitrary cutoffs. In addition, there is the initial cutoff that transforms continuous measurements (e.g., expression correlations) into unweighted edges. The ad hoc nature of these cutoffs has been a major criticism directed at this body of work. (2) These algorithms cannot be easily extended to weighted networks. Most graph-based approaches to multiple network analysis are restricted to unweighted networks, partly because weighted networks are often perceived as harder to analyze [36]. However, weighted networks are obviously more informative than their unweighted counterparts. Generating an unweighted network by applying a threshold to weighted edges invariably leads to information loss [41]. Furthermore, if there is no reasonable way to choose the threshold, this loss cannot be controlled. Both problems justify the development of an efficient computational framework suitable for mining patterns in many large weighted networks.

Generally speaking, a network of n vertices can be represented as $n \times n$ adjacency matrix $A = (a_{ij})_{n \times n}$, where each element a_{ij} is the weight of the edge between vertices i and j. A number of numerical methods for matrix computation have been elegantly applied to network analysis, for example, graph clustering [12, 15, 31, 37] and pathway analysis [5, 6]. In light of these successful applications, we propose a tensor-based computational framework capable of analyzing multiple weighted and unweighted networks in an efficient, effective, and scalable manner.

Simply put, a tensor is a multi-dimensional array and a matrix is a second-order tensor. Given m networks with the same n vertices but different topologies, we can represent the whole system as a third-order tensor $\mathscr{A} = (a_{ijk})_{n \times n \times m}$. Each element a_{ijk} is the weight of the edge between vertices i and j in the kth network. By representing a set of networks in this fashion, we gain access to a wealth of numerical methods – in particular continuous optimization methods. In fact, reformulating discrete problems as continuous optimization problems is a long-standing tradition in graph theory. There have been many successful examples, such as using a Hopfield neural network for the traveling salesman problem [22] and applying the Motzkin–Straus theorem to solve the clique-finding problem [35].

Continuous optimization techniques offer several advantages over discrete pattern mining methods. First, we may discover unexpected theoretical properties that would be invisible in a purely discrete analysis. For example, Motzkin and Straus's continuous formulation of the clique-finding problem revealed some remarkable and intriguing properties of cliques which directly benefit this work. Second, when a graph pattern mining problem is transformed into a continuous optimization problem, it becomes easy to incorporate constraints representing prior knowledge. Finally, advanced continuous optimization techniques require very few ad hoc parameters. Although tensor analysis has been productively applied in the fields of psychometrics [11, 49], image processing and computer vision [3, 48], chemometrics [43], and social network analysis [1, 27], this approach has been explored only recently in large-scale data mining [17, 32, 45–47] and bioinformatics [2, 4, 38].

In this section, we develop a tensor-based computational framework to analyze multiple weighted networks by generalizing the problem of finding heavy subgraphs in a single weighted network. A *heavy subgraph* (HS) is a subset of nodes which are heavily interconnected. We extend this concept to multiple weighted networks. By defining a *recurrent heavy subgraph* (RHS) as a subset of nodes which are heavily interconnected in a subset of weighted networks with identical nodes but different topologies. A RHS can be intuitively understood as HS that appears in multiple networks. The nodes of the RHS are always the same, although the weights of the edges may vary between networks (Fig. 6.7).

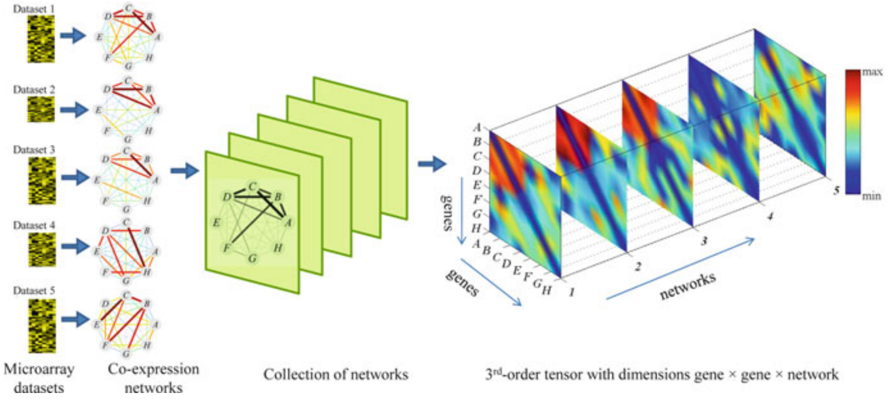


Fig. 6.7 A collection of coexpression networks can be “stacked” together into a third-order tensor such that each slice represents the adjacency matrix of one network. The weights of edges in the coexpression networks and their corresponding tensor elements are indicated by the color scale to the right of the figure. After reordering the tensor using the gene and network membership vectors, it becomes clear that the subtensor in the *top-left* corner of the tensor (formed by genes A, B, C, D in networks $1, 2, 3$) corresponds to a recurring heavy subgraph

4.1 Problem Formulation and Optimization Algorithm

Given m networks with the same n vertices but different topologies, we can represent the whole system as a third-order tensor $\mathscr{A} = (a_{ijk})_{n\times n\times m}$. Each element a_{ijk} is the weight of the edge between vertices i and j in the kth network. The genes and networks forming an RHS are described by two membership vectors: (1) the *gene membership vector* $\mathbf{x} = (x_1, \ldots, x_n)^T$, where $x_i = 1$ if gene i belongs to the RHS and $x_i = 0$ otherwise; and (2) the *network membership vector* $\mathbf{y} = (y_1, \ldots, y_m)^T$, where $y_j = 1$ if the RHS appears in the network j and $y_j = 0$ otherwise. The summed weight of all edges in the RHS is

$$H_{\mathscr{A}}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m} a_{ijk} x_i x_j y_k \tag{6.6}$$

Note that only the weights of edges a_{ijk} with $x_i = x_j = y_k = 1$ are counted in $H_{\mathscr{A}}$. Thus, $H_{\mathscr{A}}(\mathbf{x}, \mathbf{y})$ measures the “heaviness” of the network defined by $\mathbf{x}$ and $\mathbf{y}$.

To identify a RHS of K_1 genes and K_2 networks intuitively, we should look for the binary membership vectors $\mathbf{x}$ and $\mathbf{y}$ that jointly maximize $H_{\mathscr{A}}$ under the constraints $\sum_{i=1}^{n} x_i = K_1$ and $\sum_{j=1}^{m} y_j = K_2$. This cubic integer programming problem is NP-hard [39]. We instead seek an efficient polynomial solution by reformulating the task as a continuous optimization problem. That is, we look for real vectors $\mathbf{x}$ and $\mathbf{y}$ that jointly maximize $H_{\mathscr{A}}$. This optimization problem is formally expressed as follows:

$$\begin{array}{ll} \max_{\mathbf{x}\in\mathbb{R}_+^n,\mathbf{y}\in\mathbb{R}_+^m} & H_{\mathscr{A}}(\mathbf{x},\mathbf{y}) \\ \text{subject to} \begin{cases} f(\mathbf{x}) = 1 \\ g(\mathbf{y}) = 1 \end{cases} & , \end{array} \tag{6.7}$$

where $\mathbb{R}_+$ is a nonnegative real space, and $f(\mathbf{x})$ and $g(\mathbf{y})$ are vector norms. This formulation describes a tensor-based computational framework for the RHS identification problem. By solving (6.7), users can easily identify frequent heavy subgraphs consisting of the top-ranking networks (after sorting the tensor by $\mathbf{y}$) and top-ranking genes (after sorting each network by $\mathbf{x}$). After discovering the heaviest RHS in this manner, we can mask it with zeros and optimize (6.7) again to search for the next heaviest RHS.

Two major components of the framework described in (6.7) remain to be designed: (1) the vector norm constraints $(f(\mathbf{x}), g(\mathbf{y}))$, and (2) a protocol for maximizing $H_{\mathscr{A}}(\mathbf{x},\mathbf{y})$. We explain our design choices below.

4.1.1 Vector Norm Constraints

The choice of vector norms will significantly impact the outcome of the optimization. The norm of a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ is typically defined in the form $\|\mathbf{x}\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$, where $p \geqslant 0$. The symbol $\|\mathbf{x}\|_p$, called the "L_p-vector norm," refers to this formula for the given value of p. In general, the L_0 norm leads to sparse solutions where only a few components of the membership vectors are significantly different from zero [52]. The L_∞ norm generally gives a "smooth" solution where the elements of the optimized vector are approximately equal.

In our problem, a RHS is a subset of genes that are heavily connected to each other in as many networks as possible. These requirements can be encoded as follows. (1) *A subset of values in each gene membership vector should be significantly nonzero and close to each other, while the rest are close to zero*. To this end, we consider the mixed norm $L_{0,\infty}(\mathbf{x}) = \alpha\|\mathbf{x}\|_0 + (1-\alpha)\|\mathbf{x}\|_\infty$ $(0 < \alpha < 1)$ for $f(\mathbf{x})$. Since L_0 favors sparse vectors and L_∞ favors uniform vectors, a suitable choice of α should yield vectors with a few nonzero significant elements that are similar in magnitude, while all other elements are close to zero. In practice, we approximate $L_{0,\infty}$ with the mixed norm $L_{p,2}(\mathbf{x}) = \alpha\|\mathbf{x}\|_p + (1-\alpha)\|\mathbf{x}\|_2$, where $p < 1$. (2) *As many network membership values as possible are nonzero and close to each other*. As discussed above, this is the typical outcome of optimization using the L_∞ norm. In practice, we approximate L_∞ with $L_q(\mathbf{y})$ where $q > 1$ for $g(\mathbf{y})$. In our experiments, we tested several different settings and finally settled on $p = 0.8$, $\alpha = 0.2$, and $q = 10$ as effective choices for discovering a RHS.

4.1.2 Multi-Stage Convex Relaxation Optimization

Our tensor framework requires an effective optimization method that can deal with nonconvex constraints. It is well-known that the global optimum of a convex problem can be easily computed, while the quality of the optimum for a nonconvex problem depends heavily on the numerical procedure. Standard numerical techniques such as gradient descent lead to a local minimum of the solution space, and different procedures often find different local minima. Considering the fact that most sparse constraints are nonconvex, it is important to find a theoretically justified numerical procedure that leads to a *reproducible solution*.

We use our previously developed framework, known as Multi-Stage Convex Relaxation (MSCR) [52, 53], to design the optimization protocol. MSCR has good statistical properties, and has been proven to generate reproducible solutions even for nonconvex optimization problems [52, 53]. In this context, concave duality will be used to construct a sequence of convex relaxations that give increasingly accurate approximations to the original nonconvex problem. We approximate the sparse constraint function $f(\mathbf{x})$ by the convex function $\widetilde{f}_{\mathbf{v}}(\mathbf{x}) = \mathbf{v}^T h(\mathbf{x}) + f_h^*(\mathbf{v})$, where $h(\mathbf{x})$ is a specific convex function $h(x) = x^h$ ($h \geqslant 1$) and $f_h^*(\mathbf{v})$ is the concave dual of the function $\overline{f}_h(\mathbf{v})$ (defined as $f(\mathbf{v}) = \overline{f}_h(h(\mathbf{v}))$). The vector $\mathbf{v}$ contains coefficients that will be automatically generated during the optimization process. After each optimization, the new coefficient vector $\mathbf{v}$ yields a convex function $\widetilde{f}_{\mathbf{v}}(\mathbf{x})$ that more closely approximates the original nonconvex function $f(\mathbf{x})$.

4.2 *Experimental Study*

We applied our methods to 129 microarray datasets generated by different platforms and collected from the NCBI GEO. We used only datasets containing at least $\geqslant$20 samples, to ensure that correlations in the coexpression networks were very robust. Each microarray dataset is modeled as a coexpression graph following the method introduced in Sect. 2.1.3.

We identified 4,327 RHSs, each of which contains at least five member genes and occur in at least five networks. The minimum "heaviness" of these patterns is 0.4. The average size is 8.5 genes, and the average recurrence is 10.1 networks. To assess the quality of these RHSs, we evaluate the functional homogeneity of their member genes using both Gene Ontology Analysis and KEGG pathway analysis.

For each RHS, we test its enrichment for specific Gene Ontology (GO) biological process terms and GO cellular component terms [14]. To ensure the specificity of GO terms, we removed from consideration any terms associated with more than 500 genes. If the member genes of a RHS are enriched in a GO term with a hypergeometric p-value less than 0.001, we declare the RHS to be functionally homogeneous. Our results show that 59.7% of RHSs with $\geqslant$5 member genes, $\geqslant$5 recurrences, and $\geqslant$0.4 heaviness were functionally homogenous. To highlight the significance of this result, we generated random patterns with the same size distribution as

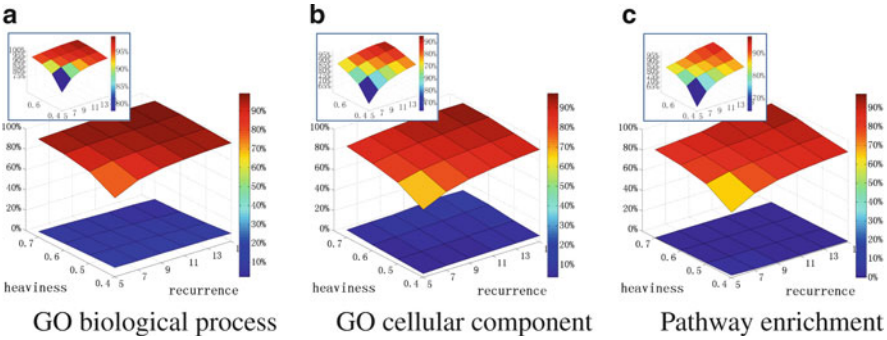


Fig. 6.8 Evaluating the functional homogeneity of RHSs using three forms of enrichment analysis. Each method is presented in two plots: the larger plot shows the difference between enrichment results on RHSs and random patterns; while the smaller plot focuses on the results of RHSs alone. It is obvious that the functional enrichments of RHSs are much greater than those found in random patterns, and also that the quality of the RHSs increases significantly with heaviness and recurrence

the RHSs. Only 9.3% of these patterns were functionally homogenous. The functionally homogenous RHSs cover a wide-range of biological processes, including translational elongation, mitosis, cell cycle, RNA splicing, ribosome biogenesis, histone modification, chromosome localization, spindle checkpoint, posttranscriptional regulation, and protein folding. Our statistical analysis also demonstrates that the greater the heaviness and recurrence, the more likely it is to be functionally homogenous. This relationship is shown in Fig. 6.8a,b.

We used KEGG human pathways[2] to assess the degree to which RHS modules represent known biological pathways. If member genes of a RHS are enriched in a pathway with a hypergeometric p-value less than 0.001, we declare the RHS to be “pathway homogeneous.” The results show that 43.5% of RHSs with $\geqslant$5 genes, $\geqslant$5 recurrences, and $\geqslant$0.4 heaviness were pathway homogenous, compared to a rate of 1.7% in randomly generated patterns (Fig. 6.8c). The RHSs are enriched in a variety of pathways: oxidative phosphorylation, cell cycle, cell communication, focal adhesion, ECM-receptor interaction, glycolysis, etc.

5 Conclusion

Biological network data are rapidly accumulating for a wide-range of organisms under various conditions. The integrative analysis of multiple biological networks is a powerful approach to discover meaningful network patterns, including subtle structures and relationships that could not be discovered in a single network.

[2] http://www.genome.jp/kegg/.

In this chapter, we proposed several novel types of recurrent patterns and derived algorithms to discover them. We also demonstrated that the identified patterns can facilitate functional discovery, regulatory network reconstruction, and phenotype characterization. Although we used coexpression networks as examples throughout this work, our methods can be applied to other types of relational graphs for pattern discovery. New challenges will arise as the quantity and complexity of biological network data continue to increase. The wealth of biological data will certainly push the scale and scope of graph-based data mining to the next level.

Acknowledgments The work presented in this chapter was supported by National Institutes of Health Grants R01GM074163, P50HG002790, and U54CA112952 and NSF Grants 0515936, 0747475 and DMS-0705312.

References

1. Acar E, Camtepe SA, Krishnamoorthy M, Yener B (2005) Modeling and multiway analysis of chatroom tensors. In: Proc of IEEE Int. Conf. on Intelligence and Security Informatics, pp 256–268
2. Acar E, Aykut-Bingol C, Bingol H, Bro R, Yener B (2007) Multiway analysis of epilepsy tensors. Bioinformatics 23(13):i10–18
3. Aja-Fernández S, de Luis García R, Tao D, Li X (eds) (2009) Tensors in Image Processing and Computer Vision. Advances in Pattern Recognition, Springer
4. Alter O, Golub GH (2005) Reconstructing the pathways of a cellular system from genome-scale signals by using matrix and tensor computations. Proc Natl Acad Sci USA 102(49):17559–17564
5. Alter O, Brown P, Botstein D (2000) Singular value decomposition for genome-wide expression data processing and modeling. Proc Natl Acad Sci USA 97(18):10101–10106
6. Alter O, Brown P, Botstein D (2003) Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms. Proc Natl Acad Sci USA 100(6):3351–3356
7. Barabasi A, Oltvai Z (2004) Network biology: understanding the cell's functional organization. Nature Reviews Genetics 5(2):101–113
8. Breiman L (2001) Random forests. Machine Learning 45(1):5–32
9. Butte AJ, Chen R (2006) Finding disease-related genomic experiments within an international repository: first steps in translational bioinformatics. AMIA Annual Symposium proceedings pp 106–110
10. Butte AJ, Kohane IS (2006) Creation and implications of a phenome-genome network. Nat Biotechnol 24(1):55–62
11. Cattell RB (1952) The three basic factor-analytic research designs-their interrelations and derivatives. Psychological Bulletin 49:499–452
12. Chung FRK (1997) Spectral Graph Theory. No. 92 in CBMS Regional Conference Series in Mathematics, American Mathematical Society
13. Collette Y, Siarry P (2003) Multiobjective Optimization: Principles and Case Studies. Springer
14. Consortium GO (2006) The gene ontology (go) project in 2006. Nucleic Acids Res 34(Database issue):D322–6
15. Ding C, He X, Zha H (2001) A spectral method to separate disconnected and nearly-disconnected web graph components. In: Proc of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM New York, NY, USA, pp 275–280

16. Edgar R, Domrachev M, Lash AE (2002) Gene expression omnibus: NCBI gene expression and hybridization array data repository. Nucleic Acids Research 30(1):207–210
17. Faloutsos C, Kolda TG, Sun J (2007) Mining large graphs and streams using matrix and tensor tools. In: Proc. of the ACM SIGMOD International Conference on Management of Data, p 1174
18. Geman S, Geman D (1984) Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6: 721–741
19. Gollub J, Ball CA, Binkley G, Demeter J, Finkelstein DB, Hebert JM, Hernandez-Boussard T, Jin H, Kaloper M, Matese JC, Schroeder M, Brown PO, Botstein D, Sherlock G (2003) The stanford microarray database: data access and quality assessment tools. Nucleic Acids Research 31(1):94–96
20. Grahne G, Zhu J (2003) Efficiently using prefix-trees in mining frequent itemsets. In: FIMI'03 Workshop on Frequent Itemset Mining Implementations
21. Hermjakob H, Montecchi-Palazzi L, Lewington C, Mudali S, Kerrien S, Orchard S, Vingron M, Roechert B, Roepstorff P, Valencia A, Margalit H, Armstrong J, Bairoch A, Cesareni G, Sherman D, Apweiler R (2004) IntAct: an open source molecular interaction database. Nucleic Acids Research 32(Database issue):D452–455
22. Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci USA 79(8):2554–2558
23. Hu H, Yan X, Huang Y, Han J, Zhou XJ (2005) Mining coherent dense subgraphs across massive biological networks for functional discovery. Bioinformatics 21(Suppl 1):i213–221
24. Huang Y, Li H, Hu H, Yan X, Waterman MS, Huang H, Zhou XJ (2007) Systematic discovery of functional modules and context-specific functional annotation of human genome. Bioinformatics 23(13):i222–229
25. Kelley B, Sharan R, Karp R, Sittler T, Root D, Stockwell B, Ideker T (2003) Conserved pathways within bacteria and yeast as revealed by global protein network alignment. Proc Natl Acad Sci USA 100(20):11394–11399
26. Kirkpatrick S, Gelatt C, Vecchi M (1983) Optimization by simulated annealing. Science 220(4598):671–680
27. Kolda TG, Bader BW, Kenny JP (2005) Higher-order web link analysis using multilinear algebra. In: Proc of IEEE Int. Conf. on Data Mining, pp 242–249
28. Koyutürk M, Grama A, Szpankowski W (2004) An efficient algorithm for detecting frequent subgraphs in biological networks. Bioinformatics 20 Suppl 1:i200–207
29. Koyutürk M, Kim Y, Subramaniam S, Szpankowski W, Grama A (2006) Detecting Conserved Interaction Patterns in Biological Networks. J Comput Biol 13(7):1299–1322
30. Koyutürk M, Kim Y, Topkara U, Subramaniam S, Szpankowski W, Grama A (2006) Pairwise alignment of protein interaction networks. J Comput Biol 13(2):182–199
31. Luxburg U (2007) A tutorial on spectral clustering. Statistics and Computing 17(4):395–416
32. Mahoney M, Maggioni M, Drineas P (2008) Tensor-CUR decompositions for tensor-based data. SIAM Journal on Matrix Analysis and Applications 30:957–987
33. Mehan MR, Nunez-Iglesias J, Kalakrishnan M, Waterman MS, Zhou XJ (2009) An integrative network approach to map the transcriptome to the phenome. J Comput Biol 16(8):1023–1034
34. Mitchell JA, Aronson AR, Mork JG, Folk LC, Humphrey SM, Ward JM (2003) Gene indexing: characterization and analysis of nlm's generifs. AMIA Annual Symposium proceedings pp 460–4
35. Motzkin TS, Straus EG (1965) Maxima for graphs and a new proof of a theorem of Turán. Canad J Math 17(4):533–540
36. Newman MEJ (2004) Analysis of weighted networks. Phys Rev E 70(5):056131
37. Ng A, Jordan M, Weiss Y (2001) On spectral clustering: Analysis and an algorithm. In: Proc. Advances in Neural Information Processing Systems, pp 849–856
38. Omberg L, Golub GH, Alter O (2007) A tensor higher-order singular value decomposition for integrative analysis of DNA microarray data from different studies. Proc Natl Acad Sci USA 104(47):18371–18376

39. Papadimitriou CH (1981) On the complexity of integer programming. Journal of the ACM 28(4):765–768
40. Papin J, Price N, Wiback S, Fell D, Palsson B (2003) Metabolic pathways in the post-genome era. Trends Biochem Sci 28(5):250–258
41. Serrano MA, Boguñá M, Vespignani A (2009) Extracting the multiscale backbone of complex weighted networks. Proc Natl Acad Sci USA 106(16):6483–6488
42. Sharan R, Suthram S, Kelley RM, Kuhn T, McCuine S, Uetz P, Sittler T, Karp RM, Ideker T (2005) Conserved patterns of protein interaction in multiple species. Proc Natl Acad Sci USA 102(6):1974–1979
43. Smilde A, Bro R, Geladi P (2004) Multi-way Analysis: Applications in the Chemical Sciences. Wiley, West Sussex, England
44. Suman B, Kumar P (2006) A survey of simulated annealing as a tool for single and multiobjective optimization. Journal of the Operational Research Society 57(10):1143–1160
45. Sun J, Tao D, Faloutsos C (2006) Beyond streams and graphs: dynamic tensor analysis. In: Proc of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 374–383
46. Sun J, Tao D, Papadimitriou S, Yu PS, Faloutsos C (2008) Incremental tensor analysis: Theory and applications. ACM Transactions on Knowledge Discovery from Data 2(3)
47. Sun J, Tsourakakis C, Hoke E, Faloutsos C, Eliassi-Rad T (2008) Two heads better than one: pattern discovery in time-evolving multi-aspect data. Data Mining and Knowledge Discovery 17(1):111–128
48. Tao D, Song M, Li X, Shen J, Sun J, Wu X, Faloutsos C, Maybank SJ (2008) Bayesian tensor approach for 3-d face modeling. IEEE Trans Circuits Syst Video Techn 18(10):1397–1410
49. Tucker LR (1966) Some mathematical notes on three-mode factor analysis. Psychometrika 31:279–311
50. Wu LF, Hughes TR, Davierwala AP, Robinson MD, Stoughton R, Altschuler SJ (2002) Large-scale prediction of saccharomyces cerevisiae gene function using overlapping transcriptional clusters. Nature Genetics 31(3):255–265
51. Yan X, Mehan MR, Huang Y, Waterman MS, Yu PS, Zhou XJ (2007) A graph-based approach to systematically reconstruct human transcriptional regulatory modules. Bioinformatics 23(13):i577–586
52. Zhang T (2008) Multi-stage convex relaxation for learning with sparse regularization. In: Proc. of Advances in Neural Information Processing Systems, pp 1929–1936
53. Zhang T (2009) Multi-stage convex relaxation for non-convex optimization. Tech. rep., Rutgers University
54. Zhou X, Kao MJ, Wong WH (2002) Transitive functional annotation by shortest-path analysis of gene expression data. Proc Natl Acad Sci USA 99(20):12,783–12,788
55. Zhou X, Kao M, Huang H, Wong A, Nunez-Iglesias J, Primig M, Aparicio O, Finch C, Morgan T, Wong W, et al (2005) Functional annotation and network reconstruction through cross-platform integration of microarray data. Nature Biotechnology 23:238–243

Chapter 7
Molecular Networks and Complex Diseases

Mehmet Koyutürk, Sinan Erten, Salim A. Chowdhury, Rod K. Nibbe, and Mark R. Chance

Abstract Many human diseases are based on a set of complex interactions among multiple genetic and environmental factors. Recent developments in biotechnology have enabled interrogation of the cell at various levels leading to many types of "omic" data that provide valuable information on these factors and their interactions. These data include (1) genomic data, which reveals possible genetic factors involved in disease, (2) transcriptomic data, which reveals changes in regulation of gene expression, and (3) proteomic data, which reveals irregularities in the amount of functional proteins in affected tissues. While these data are very useful in understanding differences between disease phenotypes, they provide information at the level of a single molecular type. To integrate these disparate data types, molecular network analysis is invaluable in uncovering the relations between disparate molecular targets and understanding disease development and progression at the systems level. This chapter provides an overview of current findings on the systems biology of human diseases in the context of molecular networks and outlines current computational approaches in network biology of human diseases.

1 Introduction

One of the major challenges in the postgenomic era is systems-level characterization of complex human diseases, that is, diseases that result from the interplay between multiple genetic and environmental factors. With significant advances in high-throughput screening technologies, it is now possible to develop computational techniques for driving studies aimed toward mechanistic understanding of disease

M. Koyutürk (✉)
Department of Electrical Engineering and Computer Science, Case Western Reserve University, 10900 Euclid Ave., Cleveland, OH 44106, USA
e-mail: mxk331@case.edu

M. Koyutürk et al. (eds.), *Functional Coherence of Molecular Networks in Bioinformatics*, DOI 10.1007/978-1-4614-0320-3_7,
© Springer Science+Business Media, LLC 2012

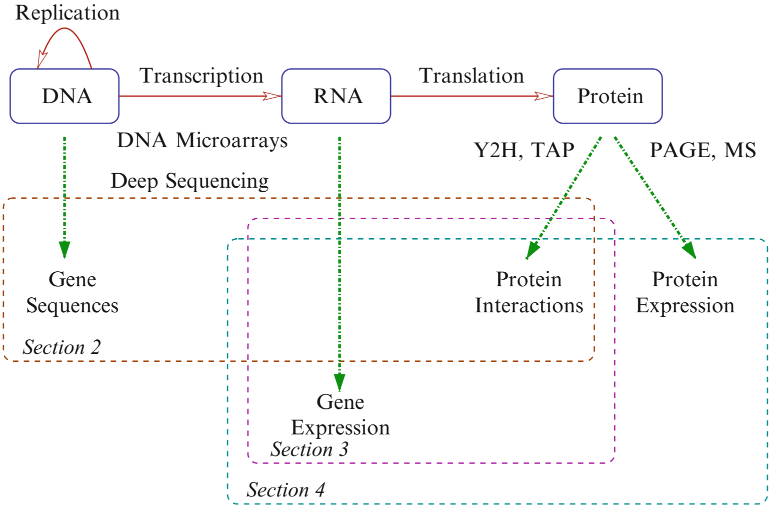


Fig. 7.1 The organization of this chapter, illustrated in the context of the central dogma of molecular biology. The composition of macromolecules in an organism can be interrogated at different levels using various technologies. Protein interaction networks lie at the core of the computational methods discussed in this chapter, and each section details current algorithmic approaches to the integration of disparate -omic datasets to analyze of complex diseases

development and progression. An important source of information that is useful in modeling functional relationships between multiple factors comes from networks of protein–protein interactions (PPIs). Challenges associated with the development of efficient computational methods for analyzing PPI networks in the context of human diseases, though, are exacerbated by the static, incomplete, and noisy nature of these network data. These challenges are further amplified by the difficulties associated with studying dynamical systems via qualitative models. Yet, novel computational methods for network-based disease analysis are rapidly emerging. These methods aim to enhance the use of various -omic datasets in understanding complex diseases, by grounding themselves on empirical evidence that provide clues on the relationship between molecular signatures of disease and functional topology of PPI networks. In this chapter, we discuss current findings on the systems biology of human diseases in the context of molecular networks and outline current computational approaches to integrating disparate -omic datasets in the study of complex diseases.

The organization of this chapter is outlined in Fig. 7.1. In Sect. 2, we discuss how genomic data is interpreted in the context of PPI networks and how network data is used to refine the findings of genome-wide linkage and association studies (GWAS). In Sect. 3, we discuss how PPI networks are used to discover network signatures of transcriptional dysregulation and how these network signatures are used to enhance diagnosis and prognosis of complex diseases. In Sect. 4, we discuss how protein expression data can be used to extract knowledge on the systems biology of complex diseases through its integration with other -omic datasets within the framework of PPI networks. Finally, in Sect. 5, we conclude with a discussion of open problems in network-based analyses of complex diseases.

2 Genomics: Prioritizing Disease Genes

Characterization of disease-associated variations in the human genome is an important step toward enhancing our understanding of the cellular mechanisms that drive complex diseases, with profound applications in modeling, diagnosis, prognosis, and therapeutic intervention [9]. Genome-wide linkage and association studies in healthy and affected populations provide chromosomal regions containing hundreds of polymorphisms that are potentially associated with certain genetic diseases [26]. These polymorphisms often span up to 300 genes, only a few of which probably have a role in the manifestation of disease. Investigation of that many candidates via sequencing is not always a feasible option. Consequently, computational methods are primarily used to prioritize and identify the most likely disease-associated genes by utilizing a variety of data sources such as gene expression [43,56] and functional annotations [1, 14, 74]. However, the scope of methods that rely on functional annotations is limited because only a small fraction of genes in the human genome are currently annotated. Moreover, signals inferred from gene expression profiles are not easily utilized, especially for diseases caused by multiple genes, where the impact of each contributor gene can be minimal, while the sum of many genes working in concert may be substantial. PPIs serve as an invaluable resource in this regard, since they provide functional information in a network context and they can be obtained at a large scale via high-throughput screening [22].

2.1 Interactions Among Disease Genes

Network-based analyses of diverse phenotypes demonstrate that products of genes that are implicated in similar diseases are clustered together into “hot spots” in PPI networks [27, 64]. Here, the similarity between diseases refers to the similarity in clinical classification of diseases. In a systematic study using the Online Mendelian Inheritance in Man (OMIM) Database, Goh et al. [27] show that the products of as many as 290 of the 903 pairs of genes that are implicated in the same disease class are also known to interact with each other ($p < 10^{-6}$). Motivated by these observations, many studies search the PPI networks for interacting partners of disease-implicated genes to narrow down the set of candidate genes implicated by GWAS [23, 35, 37, 43].

Let C denote the set of candidate genes that are within the linkage interval identified by genome wide linkage analysis for a disease of interest, D. Let S denote the set of genes that are likely to be associated with this disease based on existing knowledge. The overall objective of network-based disease gene prioritization is to use a human PPI network $G = (V,E)$, to prioritize candidate genes $c \in C$ based on their likelihood of being associated with D. Here, V denotes the set of gene products in the network and E denotes the set of interactions between these gene products, where $uv \in E$ represents an interaction between $u \in V$ and $v \in V$. In this network, the set of interacting partners of a protein $v \in V$ is denoted $N(v) = \{u \in V : uv \in E\}$.

In one of the pioneering studies on network-based disease gene prioritization, Oti et al. [57] identify potential disease genes by qualitatively investigating the interacting partners of the genes in S. Frank et al. [23] extend this idea in a quantitative framework to score genes in C based on the number of interactions between each candidate disease gene and genes likely to be associated with disease based on existing knowledge. Lage et al. [43] further refine this framework to also consider the information provided by the genes implicated in diseases similar to the disease of interest, D. In other words, rather than limiting S to the genes that are implicated in D, they also consider genes that are implicated in diseases that are "similar" to D. Here, the phenotypic similarity between two diseases is assessed in terms of the shared terms in the text and clinical synopsis parts of the OMIM record for each disease. Given these phenotypic similarity scores, they compute a score $\sigma(s,D)$ for each gene $s \in S$, which reflects the similarity between D and the diseases that are associated with s. Subsequently, the posterior probability of the disease association of each candidate gene is computed using Bayes' rule, based on the level of association between the interacting partners of the candidate gene and disease D. This method, illustrated in Fig. 7.2, is shown to rank 298 disease-causing genes as the top candidates for 669 linkage intervals studied [43].

2.2 Information-Flow-Based Methods

While methods that consider direct interactions between disease genes and candidate genes are useful, they do not utilize knowledge of PPIs to their full potential. In particular, they do not consider interactions among proteins that are not coded by candidate genes, which might also be useful in understanding indirect, but biologically important, functional relationships between candidate genes and seed genes. For this reason, they are highly vulnerable to missing interactions. Previous work shows that proteins that do not directly interact, but share interacting partners or reside in close neighborhood of each other in the network, tend to have similar biological functions and participate in common pathways [17,58]. However, including these indirect relationships raises the problem of false positives, which are known to exist in vast amounts in high-throughput PPI data.

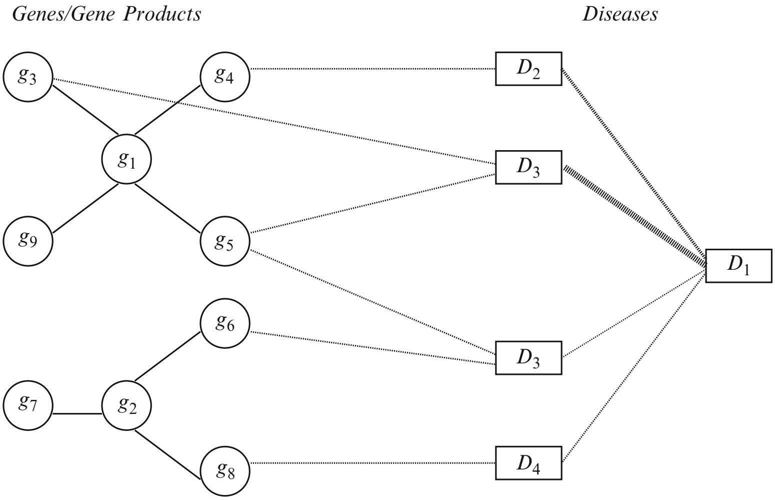


Fig. 7.2 Use of phenotypic similarity between diseases and knowledge of protein–protein interactions to prioritize candidate genes. In this example, gene products are shown by *circles*, diseases are shown by *rectangles*, available interactions between proteins are shown by *solid lines*, phenotypic similarity of diseases are shown by the *thickness* of *dashed lines*, and available gene-disease associations are shown by *dotted lines*. The disease of interest is denoted as D_1 and the candidate genes that are in the linkage interval associated with D_1 are denoted as $C = \{g_1, g_2\}$. The seed set consists of genes with disease association based on existing knowledge, that is, $S = \{g_3, g_4, g_5, g_6, g_7\}$. Based on the information shown, we can conclude that g_1 is more likely to be associated with D_1 since its product interacts with the products of genes that are implicated in diseases very similar to D_1

Information-flow-based approaches to disease gene prioritization ground themselves on the notion that products of genes that have an important role in a disease are expected to exhibit significant network crosstalk to each other in terms of the aggregate strength of paths that connect the corresponding proteins. This notion, which is illustrated in Fig. 7.3, is motivated by the following observations: (1) multiple alternate paths between functionally-associated proteins are often conserved through evolution, owing to their contribution to robustness against perturbations, as well as amplification of signals [39, 45]; (2) consideration of alternate paths accounts for missing data and noise in PPI networks [38,42]. Indeed, information-flow-based models are also shown to be very effective in network-based functional annotation of proteins [53].

2.2.1 Random Walk with Restarts

A powerful information-flow-based strategy to prioritize candidate genes is based on random walk with restarts. This method simulates a random walk on the network to compute the proximity between two nodes by exploiting the global structure

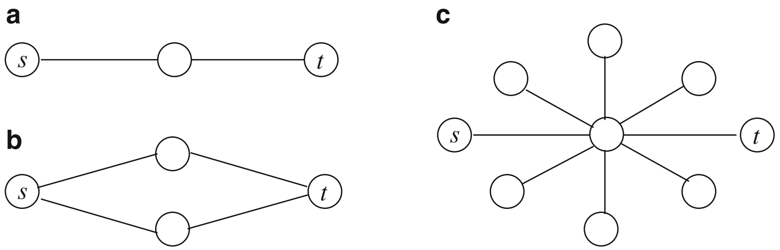


Fig. 7.3 Motivating example for consideration of multiple paths in assessing the functional association between proteins in the PPI network. Although the distance of proteins s and t is 2 in all the three hypothetical networks shown, s and t are more likely to be functionally associated in (**b**) compared to (**a**), since there are two paths connecting s and t in (**b**). On the other hand, the common neighbor in (**c**) is most likely a hub protein, thus the indirect path between s and t may not imply any functional association between s and t since hub proteins most likely interact with many proteins in many different functional contexts

of the network [47, 72]. It is used in a wide range of applications, including identification of functional modules in biological networks [48] and modeling the evolution of social networks [71]. Recently, it is also applied to candidate disease gene prioritization [13, 14, 41].

In the context of disease gene prioritization, random walk with restarts is applied as follows. A random walk starts at one of the nodes in S. At each step, the random walk either moves to a randomly chosen neighbor $u \in N$ of the current gene v or it restarts at one of the genes in the seed set S. The probability of restarting at a given time step is a fixed parameter denoted by r. For each restart, the probability of restarting at $s \in S$ is a function of $\sigma(s,D)$, that is, the degree of association between s and the disease of interest. After a sufficiently long time, the probability of being at node s at a random time step provides a measure of the association between s and the genes implicated in disease D [13,41]. Algorithmically, random-walk based association scores can be computed iteratively as follows:

$$x_{t+1} = (1-r)P_{\mathrm{RW}}x_t + r\rho. \tag{7.1}$$

Here, ρ denotes the restart vector with $\rho(s) = \sigma(s,D)/\sum_{s' \in S}\sigma(s',D)$ for $s \in S$ and 0 otherwise. P_{RW} denotes the stochastic matrix derived from G, that is, $P_{\mathrm{RW}}(u,v) = 1/|N(v)|$ for $vu \in E$ and 0 otherwise. For each $v \in V$, $x_t(v)$ denotes the probability that the random walk will be at v at time t, where $x_0 = \rho$. For each gene v, the resulting random-walk based association score is defined as $\alpha_{\mathrm{RW}}(v,D) = \lim_{t\to\infty} x_t(v)$. Finally, the candidate genes in C are ranked according to these association scores (a higher α_{RW} indicating better likelihood of being associated with the disease).

Figure 7.4 demonstrates the power of random walk with restarts in capturing the functional association between proteins in a PPI network. As seen in the figure, in comparison to shortest paths, association scores computed using random walks exhibit higher correlation with the functional similarity among proteins [58].

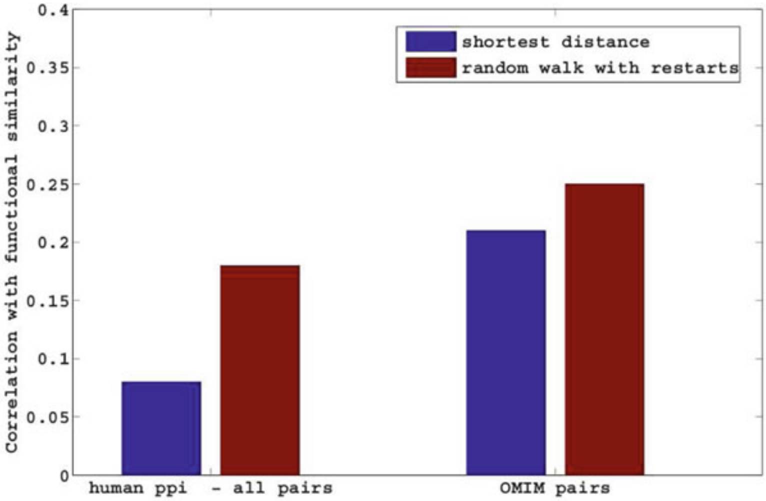


Fig. 7.4 Comparison of shortest paths and random walk with restarts in terms of their correlation with the functional association of protein pairs, as computed based on an information-theoretic measure of functional similarity using Gene Ontology [59]. OMIM pairs include all protein pairs whose coding genes are associated with the same disease whereas all pairs is the set of all other protein pairs in the PPI network

2.2.2 Network Propagation

In recent work, Vanunu et al. [77] propose a network propagation algorithm to compute the association between candidate genes and genes that are likely to be associated with disease based on existing knowledge. They define a prioritization function which models simulation of an information pump that originates at the proteins in the seed set. This idea is very similar to that of random walk with restarts, with one key difference. Namely, in network propagation, the flow of information is normalized by not only the total outgoing flow from each node, but also the total incoming flow into each node. In other words, the matrix P_{RW} is replaced by a matrix P_{NP}, in which each entry is normalized with respect to row and column sums. The resulting propagation-based model can also be simulated iteratively as follows:

$$y_{t+1} = (1-r)P_{\mathrm{NP}}y_t + r\rho. \tag{7.2}$$

Here, the propagation matrix P_{NP} is computed as $P_{\mathrm{NP}}(u,v) = 1/\sqrt{|N(u)||N(v)|}$ for $uv \in E$, 0 otherwise. For each $v \in V$, $y_t(v)$ denotes the amount of disease association information at node v at step t, where $y_0 = \rho$. For each gene v, the resulting network propagation-based association score is defined as $\alpha_{\mathrm{NP}}(v,D) = \lim_{t\to\infty} y_t(v)$. In this model, $0 \le r \le 1$ is also a user-defined parameter that is used to adjust the relative importance of prior knowledge and network topology. This method is shown to rank the true causal gene first for 34% of the 1,369 diseases in OMIM [77].

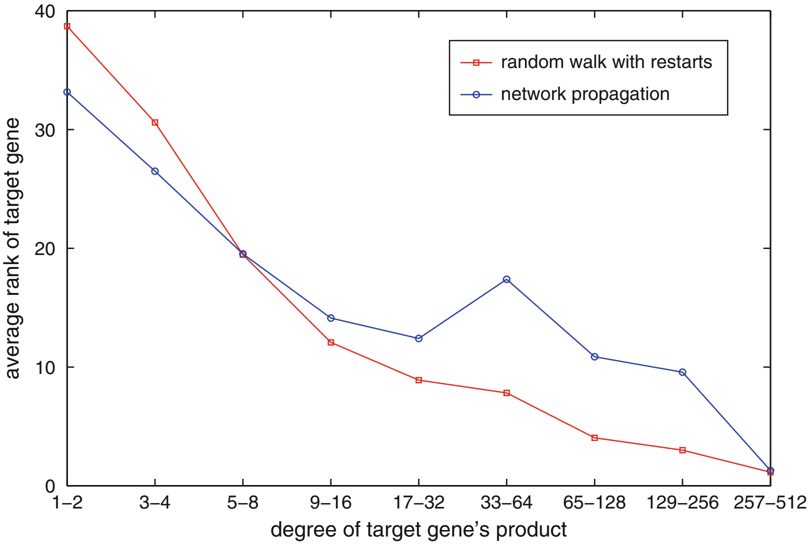


Fig. 7.5 The performance of information-flow-based methods depends on the number of available interactions of the product of true disease gene. *x*-axis represents number of interactions of the true disease gene, *y*-axis represents the average rank of true disease genes with the corresponding degree, across all disease-gene pairs in OMIM

2.3 Role of Network Centrality

While being quite powerful in capturing network-based functional association among proteins, information-flow-based methods are biased toward favoring highly connected proteins. This observation is demonstrated in Fig. 7.5. In this figure, the performance measure is the average rank of the true candidate protein (the *target protein*) among other 99 candidate proteins (e.g., proteins whose coding gene is in the same linkage interval with that of the target protein). As evident in the figure, information-flow-based methods work very well in predicting highly connected proteins, whereas they perform quite poorly for loosely connected proteins.

The dependency of performance on network degree can be understood by carefully inspecting the formulation of random walk and network propagation models. Random walk with restarts is actually a generalization of Google's well-known page-rank algorithm [8]. Indeed, for $r = 0$, α is solely a measure of network centrality. Therefore, for any $r > 0$, $\alpha(v,D)$ contains a component that represents the network centrality of v, in addition to its association with D. Network propagation alleviates this problem by normalizing the incoming flow into a gene, therefore, provides a slightly more balanced performance compared to random walk with restarts. However, as evident in the figure, its performance is still influenced heavily by node degrees. Motivated by these insights, Erten and Koyutürk [21] propose statistical correction schemes that adjust the association scores computed

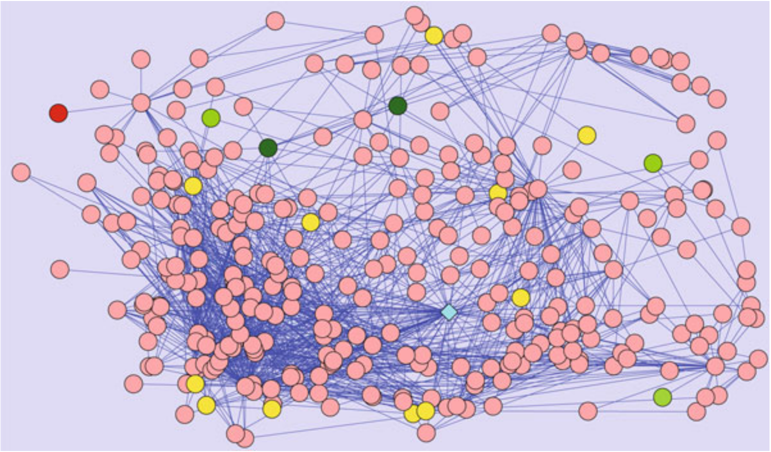


Fig. 7.6 Demonstration of the power of statistical correction in improving the performance of network-based disease gene prioritization algorithms. In this example, the disease of interest is Microphthalmia. Products of genes associated with Microphthalmia or a similar disease are shown by *green circles*, where the intensity of green is proportional to the degree of similarity. The true disease gene that is left out in the experiment and correctly ranked first by DADA [21] is represented by a *red circle*. The gene that is incorrectly ranked first for random walk with restarts and network propagation is shown by a *diamond*. Other candidate genes that are prioritized are shown by *yellow circles*

by information-flow-based algorithms based on a reference model that takes into account the degree distribution of seed and candidate proteins. As demonstrated in Fig. 7.6, the resulting algorithm, DADA (available at http://compbio.case.edu/dada/), greatly improves the performance of information-flow-based algorithms for disease gene prioritization. The case example in the figure focuses on *Microphthalmia*, which has three genes directly associated with it in OMIM; namely *SIX6*, *CHX10*, and *BCOR*. The figure shows the neighborhood up to two nodes away of the products of *SIX6*, *CHX10*, and *BCOR*. In the experiment reported, *SIX6* is removed and network-based algorithms are used to predict the true disease gene based on the network connectivity of candidate genes to *CHX10* and *BCOR*, as well as genes associated with diseases similar to Microphthalmia. As seen in the figure, the random walk with restarts and network propagation methods fail to rank *SIX6* as the first gene because the product of *SIX6* is not a centralized protein (it has only one known interacting partner in the PPI data used here). Thus, random walk with restarts ranks this true disease gene as 26th and network propagation ranks it 16th among 100 candidates. On the other hand, after statistical correction of random-walk based score with respect to network degree, DADA correctly ranks this gene as the top candidate. Both random walk and network propagation rank the gene *AKT1* top among all candidates, which not surprisingly, is a high degree protein (78 known interactions), also connected to other hub proteins.

2.4 Mechanistic Bases of Genetic Effects

Besides their use in enhancing discovery of novel disease genes, network models are also useful in understanding the mechanistic bases of observed genotype-phenotype relationships. Kelley and Ideker [39] demonstrate the use of PPI networks in explaining genetic interactions by identifying network patterns that correspond to *within-pathway* and *between-pathway* models of synthetic lethal interactions in yeast. Namely, within-pathway models refer to the case where the proteins coded by interacting genes also interact with each other physically. On the other hand, between-pathway models refer to the case where two groups of physically interacting proteins are linked to each other through genetic interactions between their coding genes. Systematic analyses of yeast synthetic lethal interactions and the yeast PPI network show that both within-pathway and between-pathway models can explain different genetic interactions.

Recently, in the context of human diseases, systematic computational methods are proposed to identify pathways that can (1) suggest a compact set of mutations functionally linked to the disease and (2) provide mechanistic models of the effect of genetic mutations on the development of disease. In one of these methods, Vandin et al. [76] use an information-flow-based algorithm to detect significantly mutated pathways in cancer. Based on an idea similar to that in disease gene prioritization, this method aims to utilize network information to overcome the computational challenges posed by the heterogeneous nature of somatic mutations in many cancers. Similarly, Kim et al. [40] simultaneously identify causal genes and dysregulated pathways in complex disease by integrating copy number variation and gene expression data within the framework of PPI networks, using an integer programming formulation based on a model of current flow through the PPI network.

In the context of understanding specific human diseases, more detailed studies focus on specific interactions and make use of human PPI data to derive a network model of the crosstalk between genetically interacting proteins. For example, Patel et al. [60] focus on the interaction between *APC* and *CDKN1A* in development of colorectal cancer, which are known to be synergistic in tumorigenesis in mouse models. By integrating human PPI data with other sources of -omic datasets, Patel et al. identify a network of PPIs that connects *APC* and *CDKN1A*. Subsequently, using measurements of protein and gene expression changes in intestinal epithelial tissue of mice mutated individually at *APC* (Apc1638N$+/-$) or *CDKN1A* (Cdkn1a$-/-$), they show that the predicted *APC–CDKN1A* network is significantly perturbed at the mRNA-level by both single gene knockouts.

3 Transcriptomics: Discovering Dysregulated Subnetworks

Interrogation of genomic sequences in healthy and affected populations highlight genetic variation that is potentially related to disease. However, to have a comprehensive understanding of the relationship between genotype and phenotype, it is also useful to investigate how gene expression is affected during development and progression of disease. Indeed, in the past decade, genome-wide monitoring of gene expression, enabled by DNA microarray technology, has been commonly used to interrogate the development and progress of complex diseases [69]. Today, gene expression can be measured more reliably using whole transcriptome shotgun sequencing (RNAseq) [80].

Differential analysis of gene expression facilitates identification of genes that are *dysregulated* with respect to the disease of interest; that is, genes that exhibit significant difference in the amount of mRNA transcripts present in a range of disease and control samples (e.g., samples taken from cancerous vs normal tissues). To date, systematic analyses of differential gene expression has led to identification of genetic markers associated with many complex diseases, including leukemia [28], lymphoma [2], breast cancer [62], lung cancer [6], and prostate cancer [44]. However, univariate analysis of differential expression has limited ability in uncovering the mechanistic bases of complex diseases, since many complex diseases result from the interplay among multiple interacting factors. To this end, molecular networks prove invaluable in identifying groups of interacting proteins that are coordinately dysregulated at the mRNA-level.

3.1 Integration of Gene Expression and Molecular Network Data

Molecular networks provide static and qualitative descriptions of the wiring of cellular systems. Molecular expression data, on the other hand, provides quantitative information on the molecular composition of the system under different conditions/samples, or over time. Consequently, it is natural to integrate these two sources of data to gain insights on the dynamic organization of cellular systems. Indeed, identification of groups of molecules with correlated expression profiles and coherent network connectivity patterns is shown to enhance modularization of networks [52, 78]. These approaches generally search for subnetworks with high connectivity (for functional modules) [31, 67, 73, 75] or linear chains of interactions/reactions (for signaling pathways) [4, 49, 63].

3.2 From Dysregulated Genes to Dysregulated Subnetworks

In the context of human diseases (or more generally phenotypic differences), systematic studies of differential gene expression in certain phenotype classes show that genes that are dysregulated with respect to the same phenotype likely interact with each other in molecular networks [51, 64]. Motivated by this observation, Ideker et al. [36] identify dysregulated subnetworks by searching for connected subgraphs of the PPI network with high aggregate significance of differential mRNA expression. Namely, for a given PPI network $G = (V, E)$, let $E_i(j)$ denote the expression of the gene coding for protein $g_i \in V$ in sample j. Assume that gene expression data from m samples are available and $C_j = 1$ indicates that sample j is a phenotype sample (e.g., taken from tumor tissue), while $C_j = 0$ indicates that sample j is a control sample (e.g., taken from a normal tissue). Let z_i denote the z-statistic of the differential expression of gene $g_i \in V$ (i.e., z_i is the standardized difference between the mean expression of g_i in phenotype and control samples). Then, the dysregulation of a subnetwork $S \subseteq V$ is defined as

$$\Delta_{\mathrm{U}}(S) = \sum_{g_i \in S} \frac{z_i}{\sqrt{|S|}}. \tag{7.3}$$

Variations of this method are shown to be effective in identifying multiple genetic markers in prostate cancer [30], melanoma [19], diabetes [46], and others [10, 61, 66]. However, these approaches are still limited in capturing the coordination in the dysregulation of multiple genes, since they assess differential expression individually for each gene. In other words, they cannot identify interacting genes that do not exhibit significant differential expression when considered individually, but exhibit significant differential expression when considered together.

3.3 Additive Coordinate Dysregulation

Chuang et al. [18] propose a multivariate formulation of subnetwork dysregulation in the context of breast cancer metastasis. For this purpose, they introduce the notion of *subnetwork activity*, defined as the aggregate expression of gene products in the subnetwork in each sample, that is, the activity of subnetwork $S \subseteq V$ is defined as

$$\overline{E}(S) = \sum_{g_i \in S} \frac{E_i}{\sqrt{|S|}}. \tag{7.4}$$

Assessment of differential expression with respect to subnetwork activity enables identification of subnetworks that are coordinately dysregulated at a sample-specific resolution; that is, groups of interacting proteins with collective mRNA-level differential expression. Namely, Chuang et al. define the dysregulation of subnetwork S as

$$\Delta_{\mathrm{A}}(S) = I(\overline{E}(S);C) = H(C) - H(C|\overline{E}(S)). \tag{7.5}$$

Here, $I(\overline{E}(S);C)$ denotes the mutual information of phenotype C and subnetwork activity of C. In other words it is the reduction in the uncertainty of phenotype upon observation of the aggregate expression of the genes in S. Uncertainty of phenotype is quantified by entropy $H(C)$ and the uncertainty after observation of subnetwork activity is quantified by conditional entropy $H(C|\overline{E}(S))$. In this chapter, we refer to Δ_{A} as *additive coordinate dysregulation* as it is based on additive assessment of the coordination between multiple genes. Subsequently, Chuang et al. identify subnetworks of the human PPI network that maximize $\Delta_{\mathrm{A}}(S)$ using a greedy algorithm and use subnetwork activity as features for classification. As compared to single gene markers, these subnetwork markers are shown to provide better classification performance in the predicting metastasis of breast cancer [18].

3.4 Coordinate Dysregulation and Cover

Additive coordinate dysregulation of a subnetwork can also be formulated in terms of the number of phenotype and control samples discriminated by the genes in that subnetwork (referred to as the *cover* of the subnetwork) [15]. To see the relationship between coordinate dysregulation and cover, consider gene expression data from paired samples. A gene g_i is said to *cover* a sample s_j *positively/negatively* if it is up-regulated/down-regulated in the phenotype sample with respect to control (e.g., $\hat{E}_i(j) = H$ and $\hat{E}_i(j') = \mathtt{L}$, where $\hat{E}_i(j)$ represents binarized expression of gene i in sample j, H represents high expression, L represents low expression and j' denotes the control sample that is paired with phenotype sample j.) Subsequently, the set of samples that are covered positively/negatively by g_i is called the *positive/negative cover set* of g_i and denoted $\mathscr{P}_i/\mathscr{N}_i$. The concept of cover is illustrated in Fig. 7.7. In this figure, $\mathscr{P}_i = \{s_1, s_2\}$, that is, g_1 covers samples s_1 and s_2 positively, since it is up-regulated in the phenotype samples compared to the control samples. Based on this formulation, it can be shown that $\Delta_{\mathrm{A}}(\{g_i\})$ is a monotonically increasing function of $|\mathscr{P}_i| - |\mathscr{N}_i|$ [15].

Observe in Fig. 7.7 that, since g_1 is dysregulated with respect to samples s_1 and s_2, it can be used to distinguish phenotype and control samples based on its expression in a given sample. However, clearly, the statistical power (or reliability) of g_1 in distinguishing phenotype and control samples depends on the number of samples that it covers. These observations suggest that interacting genes that distinguish different sets of samples may complement each other in distinguishing phenotype and control. In addition, the dysregulation of genes involved in similar processes may have similar effects on phenotype. Furthermore, since such genes are expected to be functionally related, they are likely to be in close proximity of each other in a network of interactions. Motivated by these observations, the algorithm NETCOVER [15] searches for subnetworks composed of genes that together cover all samples consistently (i.e., either positively or negatively). Here, for a given subnetwork $S \subseteq V$, the *positive* and *negative* cover sets of S are, respectively, defined as

$$\mathscr{P}(S) = \bigcup_{g_i \in S} \mathscr{P}_i \text{ and } \mathscr{N}(S) = \bigcup_{g_i \in S} \mathscr{N}_i. \tag{7.6}$$

In the context of human colorectal cancer, NETCOVER is shown to identify subnetworks that provide better classification accuracy than subnetworks identified by a greedy algorithm that aims to explicitly maximize additive coordinate dysregulation [15]. Other cover-based approaches to dysregulated subnetwork discovery are also shown to be effective in discovering network markers of various phenotypes, including Huntington's disease and breast cancer [75].

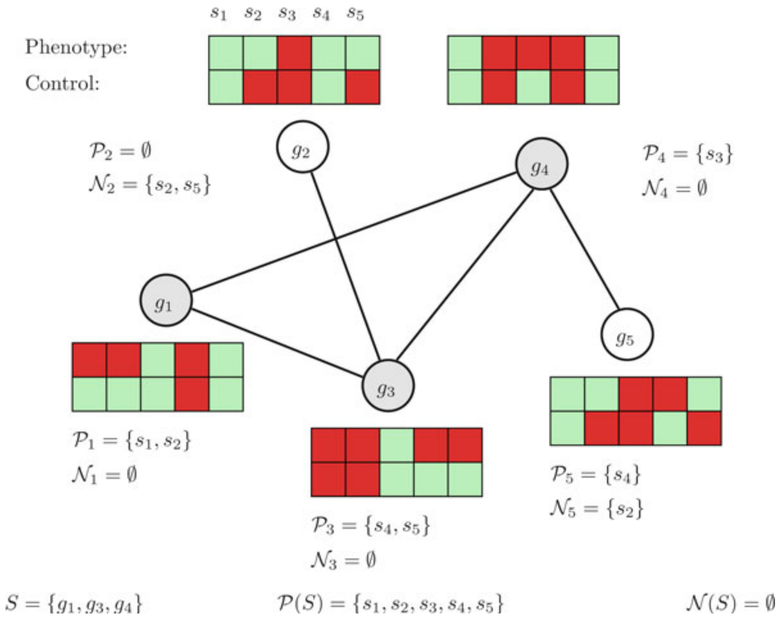

s_1 s_2 s_3 s_4 s_5
Phenotype:
Control:
g_1
g_2
g_3
g_4
g_5
$\mathcal{P}_2 = \emptyset$
$\mathcal{N}_2 = \{s_2, s_5\}$
$\mathcal{P}_4 = \{s_3\}$
$\mathcal{N}_4 = \emptyset$
$\mathcal{P}_1 = \{s_1, s_2\}$
$\mathcal{N}_1 = \emptyset$
$\mathcal{P}_3 = \{s_4, s_5\}$
$\mathcal{N}_3 = \emptyset$
$\mathcal{P}_5 = \{s_4\}$
$\mathcal{N}_5 = \{s_2\}$
$S = \{g_1, g_3, g_4\}$
$\mathcal{P}(S) = \{s_1, s_2, s_3, s_4, s_5\}$
$\mathcal{N}(S) = \emptyset$

Fig. 7.7 Cover-based formulation of coordinate dysregulation in complex phenotypes. Proteins are shown by *circles*, interaction between proteins are shown by lines connecting interacting proteins. Matrices near each protein shows its mRNA-level expression in phenotype (*upper row*) and control (*lower row*) samples from s_1 to s_5. *Dark red* indicates high expression, *light green* indicates low expression. The subnetwork composed of the grey proteins covers all samples positively

3.5 Synergistic Dysregulation

Anastassiou [3] further delineates the concept of coordinate dysregulation by defining synergy of genes based on their collective differential expression in complex phenotypes. For a given pair of genes g_i and g_j, the synergistic dysregulation of g_i and g_j is defined as

$$\Delta_S(\{g_i, g_j\}) = I(\{\hat{E}_i, \hat{E}_j\}; C) - (I(\hat{E}_i; C) + I(\hat{E}_j; C)). \tag{7.7}$$

Observe that, synergistic definition has two key differences from additive coordinate dysregulation. First, the dysregulation of the subnetwork composed of g_i and g_j is quantified in terms of the mutual information phenotype and the binary *expression state* of the subnetwork, which is a two-dimensional binary vector. This is in contrast to additive coordinate dysregulation, which quantifies coordinate dysregulation in terms of the mutual information between phenotype and average expression of the genes in the subnetwork. Second, the dysregulation of each individual gene is subtracted from the overall dysregulation of the subnetwork, so as to capture the ability of the pair in distinguishing phenotype and control beyond what the individual genes can provide. In this respect, if two genes are differentially expressed in the same manner in phenotype samples, then their synergistic dysregulation will be negative. This is because the mutual information between phenotype and expression state of the subnetwork will be equal to the mutual information between phenotype and the expression of each individual gene. On the other hand, if the two genes are complementary of each other in distinguishing phenotype and control, that is, if $I(\{\hat{E}_i,\hat{E}_j\};C) > I(\hat{E}_i;C)$ and $I(\{\hat{E}_i,\hat{E}_j\};C) > I(\hat{E}_j;C)$, then synergistic dysregulation of g_i and g_j will be positive. Therefore, the concept of synergy provides a measure for quantifying the complementarity and redundancy of two genes in distinguishing phenotype and control.

The concept of synergistic dysregulation can be extended to a subnetwork composed of multiple genes, by defining the expression state of subnetwork S of m genes. In this formulation, the expression state $F_S = \{\hat{E}_1,\hat{E}_2,\ldots,\hat{E}_m\} \in \{\mathtt{L},\mathtt{H}\}^m$ is the random variable that represents the combination of binary expression states of the genes in S. However, the number of operations required to compute the synergy of m genes is exponential in m, since one has to consider all subsets of S to compute the synergy of S. Consequently, computation of synergy for a given subnetwork of arbitrary size becomes an intractable problem, let alone the problem of identifying subnetworks with high synergy. For this reason, Watkinson et al. [81] focus on systematically identifying synergistic pairs of genes and constructing a *synergy network* by representing identified synergistic relationships as interactions between genes.

3.6 Combinatorial Coordinate Dysregulation

To overcome the computational difficulties in computing the dysregulation but still capture the combinatorial relationship of the dysregulation of multiple genes in a subnetwork, Chowdhury et al. [16] define *combinatorial coordinate dysregulation* of a subnetwork S as follows:

$$\Delta_{\mathrm{C}}(S) = I(F_S;C) = H(C) - H(C|\hat{E}_1,\hat{E}_2,\ldots,\hat{E}_m). \tag{7.8}$$

The difference between synergistic and combinatorial dysregulation is that combinatorial dysregulation does not account for the dysregulation of parts of the subnetwork; it rather aims to identify subnetworks with expression states that can distinguish phenotype and control regardless of whether or not the expression states of parts of the subnetwork can distinguish phenotype and control. The difference between additive and combinatorial coordinate dysregulation is illustrated in Fig. 7.8.

While computation of combinatorial coordinate dysregulation is straightforward, identification of subnetworks with high combinatorial coordinate dysregulation is

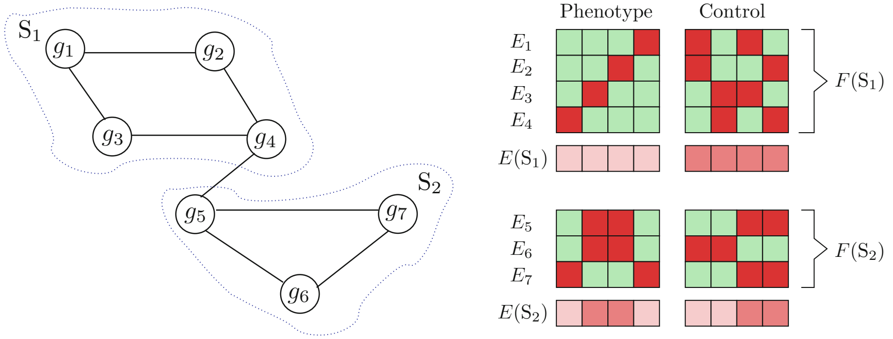


Fig. 7.8 Additive vs. combinatorial coordinate dysregulation. Genes (g) are shown as nodes, interactions between their products are shown as edges. Expression profiles (E) of genes are shown by colormaps. *Dark red* indicates high expression (H), *light green* indicates low expression (L). None of the genes can differentiate phenotype and control samples individually. Aggregate *subnetwork activity* (average expression) for each subnetwork is shown in the row below its gene expression matrix. The aggregate activity of S_1 can perfectly discriminate phenotype and control, but the aggregate activity of S_2 cannot discriminate at all. For each subnetwork S_1 and S_2, each column of the gene expression matrix specifies the *subnetwork state* in the corresponding sample. The states of both subnetworks can perfectly discriminate phenotype and control (for S_2, upregulation of g_7 alone or g_5 and g_6 together indicates phenotype; we say *state functions* LLH and HHL are indicative of phenotype)

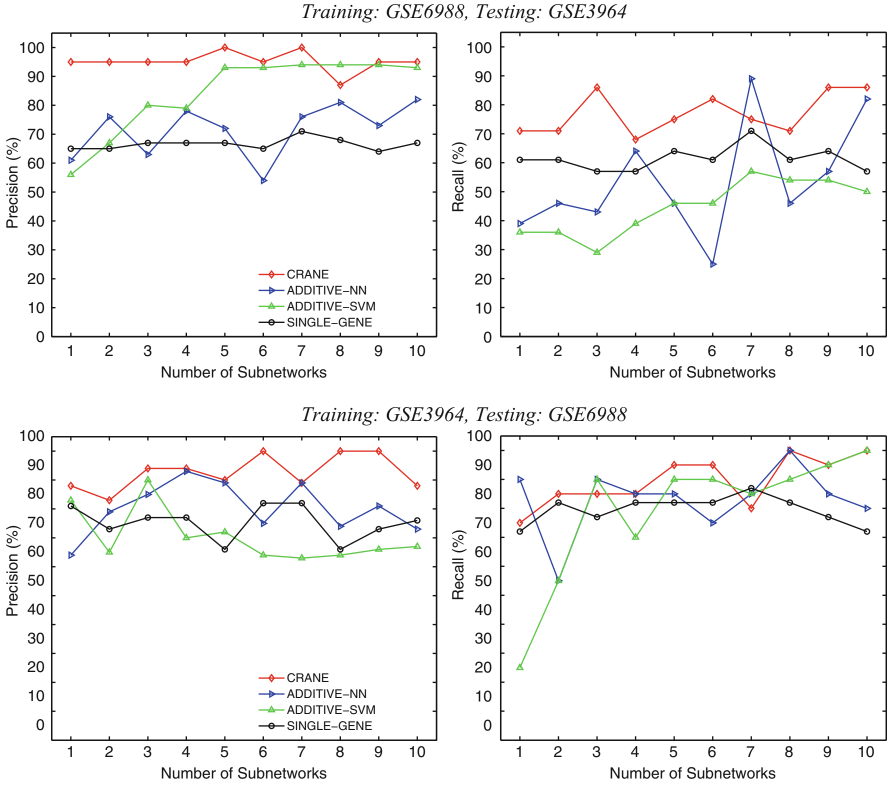
still intractable. Motivated by this consideration, Chowdhury et al. decompose the combinatorial coordinate dysregulation of a subnetwork into individual subnetwork state functions by defining

$$J(f_S;C) = p(f_S) \sum_{c \in \{0,1\}} p(c|f_S) \log(p(c|f_S)/p(c)), \tag{7.9}$$

where

$$I(F_S;C) = \sum_{f_S \in \{\mathtt{H},\mathtt{L}\}^m} J(f_S;C). \tag{7.10}$$

Here, $f_S \in \{\mathtt{H},\mathtt{L}\}^m$ denotes an observation of the random variable F_S, that is, a specific combination of the expression states of the genes in S and $p(x)$ denotes $P(X=x)$, that is, the probability that random variable X is equal to x (similarly, $p(x|y)$ denotes $P(X = x|Y = y)$). In biological terms, $J(f_S;C)$ can be considered a measure of the information provided by subnetwork *state function* f_S on phenotype C. Based on this definition, Chowdhury et al. develop an exhaustive, yet efficient algorithm, CRANE, for identification of subnetworks and associated state functions informative of phenotype (i.e., with high $J(f_S;C)$). Subsequently, they train neural networks to use identified subnetworks for classification. The performance of subnetworks identified by CRANE in predicting colon cancer metastasis is shown in Fig. 7.9.


Training: GSE6988, Testing: GSE3964

Training: GSE3964, Testing: GSE6988

Fig. 7.9 Classification performance of subnetworks identified by CRANE in predicting colon cancer metastasis, as compared to single gene markers and subnetworks identified by algorithms that aim to maximize additive coordinate dysregulation. Subnetworks identified by CRANE are used to train neural networks (NNs), while those identified by the additive algorithm are used to train NNs, as well as support vector machines (SVMs). In the graphs, horizontal axes show the number of disjoint subnetwork features (with maximum combinatorial or additive coordinate dysregulation) used in classification, vertical axes show the precision and recall achieved by the classifier

3.7 Subnetwork Markers Generate Novel Biological Insights

The power of network-based approaches in generating novel biological insights is illustrated in Fig. 7.10. The figure displays a subnetwork identified by CRANE as a subnetwork informative of metastasis in colorectal cancer. This subnetwork contains TNFSF11, MMP1, BCAN, MMP2, TBSH1, and SPP1 and the state function LLLLLH (in respective order) indicates metastatic phenotype with J-value 0.33. The combinatorial dysregulation of this subnetwork is 0.72, while its additive coordinate dysregulation is 0.37, that is, this is a subnetwork which would likely have escaped detection by the additive algorithm. Using the proteins in this

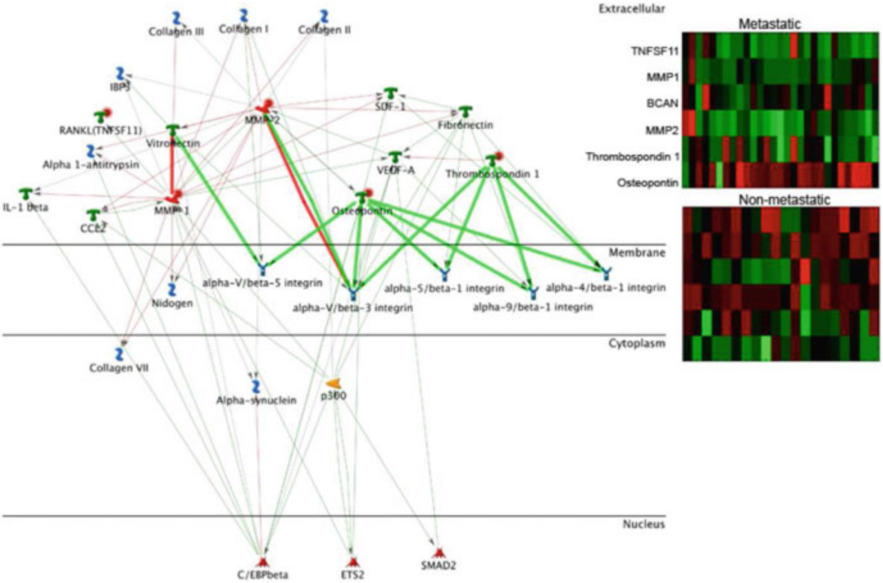


Fig. 7.10 Hypothesis-driver subnetwork – interaction diagram illustrating key interactions with gene products from a subnetwork identified by CRANE as indicative of CRC metastasis. Shown are the gene products in discovered subnetwork (*red circles*) and their direct interactions with other proteins. *Green lines* represent an activating interaction, *red lines* indicate an inhibitory interaction. *Arrows* indicate direction of interaction. Inset is the expression pattern of subnetwork proteins at the level of mRNA

subnetwork as a seed, a well-annotated subnetwork in the neighborhood of these proteins is created, with a view to more closely analyzing the post-translational interactions involving these proteins. This is accomplished using Metacore, a commercial platform that provides curated, highly reliable interactions. As seen on the interaction diagram, SPP1 (Osteopontin) and TBSH1 (Thrombosponidin 1) interact with a number of the integrin heterodimers to increase their activity (green line). Integrin heterodimers play a major role in mediating cell adhesion and cell motility. SPP1, up-regulated in metastasis, is a well-studied protein that triggers intracellular signaling cascades upon binding with various integrin heterodimers, promotes cell migration when it binds CD44, and when binding the alpha-5/beta-3 dimer in particular, promotes angiogenesis, which is associated with the metastatic phenotype of many cancers [50]. MMP proteins are involved in the breakdown of ECM, particularly collagen which is the primary substrate at the invasive edge of colorectal tumors [79]. MMP-1 has an inhibitory effect on Vitronectin (red line), hence, the loss of expression of MMP-1 may "release the brake" on Vitronectin, which in turn may increase the activity of the alpha-v/beta-5 integrin heterodimer. Likewise, MMP-2 shows an inhibitory interaction with the alpha-5/beta-3 dimer, which may counteract to some extent the activating potential of SPP1, suggesting that a loss of MMP-2 may exacerbate the metastatic phenotype. Taken together, these interactions suggest a number of perturbation experiments, perhaps by pharmacological inhibition or siRNA interference of the integrin dimmers or MMP proteins, to evaluate the role of these interactions, individually or synergistically, in maintaining the metastatic phenotype. Note also that, alpha-v/beta-5 integrin does not exhibit significant differential expression at the mRNA-level, suggesting that the state function identified by CRANE may be a signature of its post-translational dysregulation in metastatic cells.

4 Proteomics: Using Protein Expression Data Beyond Its Scale

As discussed in the previous sections, high-throughput biological data that relate to different aspects of cellular processes enable detailed studies of the interplay among multiple genetic and epigenetic factors that lead to complex diseases. At the genomic level, GWAS provide insights on the interactions between multiple genetic factors that underlie complex phenotypes [34]. At the functional level, genome-wide assays of mRNA expression enable identification of gene targets that are dysregulated with respect to phenotypes of interest, through cluster analysis, classification, differential expression, and gene selection [69]. Proteomic measurements capture functional activity more accurately, since they provide information at the post-translational level [25]. However, established proteomic screening techniques (e.g., 2D-PAGE, mass spectrometry) can only monitor the expression of a limited subset of proteins in the cell at a time. For this reason, to utilize this valuable source of information at genomic scale, it is very useful to integrate proteomic data with other, more comprehensive, data sets. Molecular networks provide an excellent resource that is structurally and functionally suited for this purpose.

4.1 Transcriptomic vs. Proteomic Data

Genome-wide screening of mRNA expression (i.e., transcriptome), enabled by DNA microarray technology and more recently by deep sequencing, is commonly utilized in identification of molecular signatures of complex diseases [69]. Basic computational approaches to the analysis of differential mRNA expression range from identification of differentially expressed genes to classification of samples for diagnostic and prognostic purposes. Furthermore, as discussed in the previous section, mRNA expression data has been increasingly employed in systems-level studies of complex phenotypes. However, it is important to note that transcriptomic data provides information on the abundance of mRNA transcripts in a sample. While this information is useful as an approximation to the abundance of functional proteins in the sample of interest, mRNA expression does not necessarily capture protein expression accurately. This is because, after transcription, protein expression is further regulated by various mechanisms of post-transcriptional modification, including alternative splicing [11], mRNA degradation [7], and RNA interference [32].

Protein expression, monitored through a variety of experimental techniques (e.g., 2D gel electrophoresis, mass spectrometry) captures post-transcriptional abundance of proteins more accurately [25]. Note that, activity of proteins is further regulated via post-translational modifications, therefore, the scope of protein expression is also limited in terms of capturing the functional activity in the cell [5]. Techniques such as flow cytometry and high-throughput phospho-proteomics offer detailed information on the functional activity of proteins [65]. However, as of today, due to various constraints on cost, scale, and practicality, they are not applied as widely as transcriptomic and proteomic screening techniques in the study of complex phenotypes.

While proteomic data proves useful in generating new insights into the systems biology of complex phenotypes [82], dedicated computational methods that utilize protein expression data are relatively scarce. This is probably because of the relatively lower coverage provided by most proteomic screening techniques, although novel proteomic methods that offer higher throughput (e.g., Reverse Phase Protein Array (RPPA) [70]) are becoming available. However, studies on understanding the relation between mRNA expression and protein expression reveal that, while transcriptional and post-translational abundance of a protein is correlated across diverse conditions, the strength of correlation is dependent on the underlying biological process [12] and may be weaker than expected for many processes [20]. Furthermore, systematic analyses of tissue-specific regulation of metabolism show that mRNA expression can account for the activity of metabolic pathways only to a certain extent, suggesting that post-translational regulation plays an important role in many metabolic processes [68]. Consequently, it has been increasingly pronounced that mRNA expression and protein expression provide information that are complementary to each other [29, 33].

4.2 Proteomics-Driven Discovery of Dysregulated Subnetworks

Based on the premise that changes in post-translational expression of a protein may be associated with synergistic changes in the transcriptional expression of a group of proteins in its neighborhood, Nibbe et al. [55] developed a proteomics-driven approach to the identification of multiple gene targets in late stages of human colorectal cancer (CRC). In this study, proteins that are differentially expressed in metastatic cells are first identified through 2D Gel analysis followed by mass spectrometry, using univariate statistics. Then, these proteins (to be precise, 67 differentially expressed proteins are discovered) are mapped on a curated human molecular network (including all types of interactions; metabolic, PPI, transcriptional, etc.) using Metacore, a commercial platform that provides curated interaction data, as well as basic features for network analysis. Subsequently, by identifying subnetworks that contain a significant number of these proteomic targets, new candidate genes in these subnetworks are selected for further transcriptomic analysis. Finally, combinations of these candidate genes are evaluated based on the significance of their coordinate mRNA-level dysregulation. For this purpose, the additive formulation of coordinate dysregulation, discussed in Sect. 3 is used [18]. The basic premise here is that, small changes in mRNA expression may lead to significant changes in the proteome. Consequently, one would expect to see significant overall transcriptomic differential expression in the network neighborhood of differentially expressed proteins. This approach is shown to be promising in identifying multiple gene targets in that its results are reproducible on different data sets.

Nibbe et al. [54] further elaborate this approach by utilizing public PPI data and developing more sophisticated algorithms and statistical models to score proteins based on their proximity to proteomic seeds. The overall proteomics-driven framework for dysregulated subnetwork discovery is shown in Fig. 7.11. As seen in the figure, the proteomics-driven procedure first identifies proteins with significant differential expression with respect to the disease, via proteomic screening. Once these targets, called *proteomic seeds*, are identified, they are mapped on the human PPI network to identify proteins that are functionally and physiologically associated with the proteomic seeds.

To score proteins in the human PPI network according to their crosstalk to proteomic seeds, Nibbe et al. use a method that is very similar to the information-flow-based method for network-based disease gene prioritization (Sect. 2). In disease gene prioritization, the seed set contains the products of genes that are likely to be associated with the disease of interest based on existing knowledge. Here, it contains the proteins that exhibit significant differential expression in the disease of interest. While the seed sets in the two problems have different biological meanings, once the seed set is fixed, the computational problem of scoring other proteins in the network is identical. Indeed, Nibbe et al. score all proteins in the network using a random-walk based model with statistical correction.

Once individual proteins are scored with respect to their crosstalk to proteomic seeds, the network is searched for implicated subnetworks; that is, subnetworks composed of proteins with (1) significant *crosstalk* to proteomic seeds and (2) coordinate/synergistic mRNA-level dysregulation with respect to the phenotype of interest. For this purpose, two types of candidate subnetworks are considered:

- *Interactor subnetworks:* For each proteomic seed, the subnetwork induced by its interacting partners in the network is considered a candidate subnetwork, based on the hypothesis that significant changes in the expression of a protein may be associated with synergistic changes in the transcriptional expression of proteins in its neighborhood.
- *Crosstalker subnetworks:* For each proteomic seed, the subnetwork induced by the proteins that interact with the seed and have significant adjusted crosstalk scores with respect to the entire proteomic seed set is considered a candidate subnetwork, based on the hypothesis that subnetworks composed of proteins with significant crosstalk to the proteomic seeds (as opposed to solely interacting with one proteomic seed) are likely to exhibit significant synergistic differential expression.

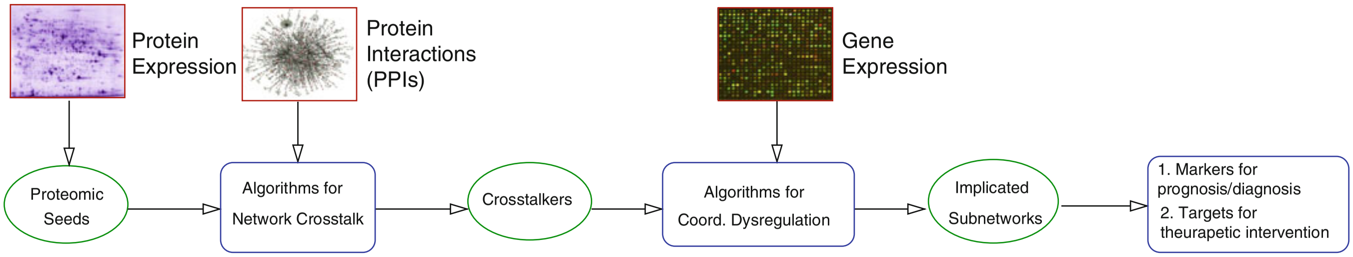


Fig. 7.11 Framework for proteomics-driven investigation of complex phenotypes

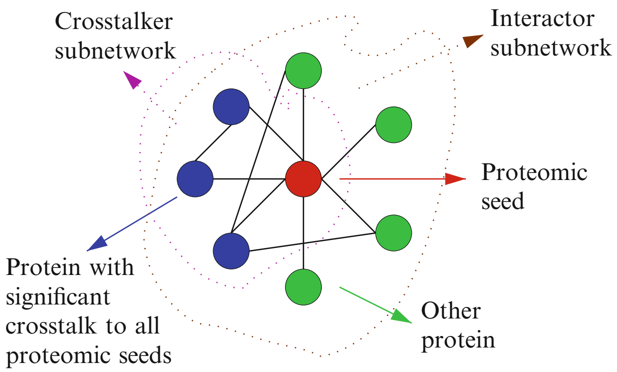


Fig. 7.12 Candidate subnetworks in proteomics-driven discovery of dysregulated subnetworks. An *interactor subnetwork* associated with a proteomic seed is composed of its interacting partners in the network. A *crosstalker subnetwork* associated with a proteomic seed, on the other hand, contains only the interacting partners that exhibit significant crosstalk to the entire set of proteomic seeds

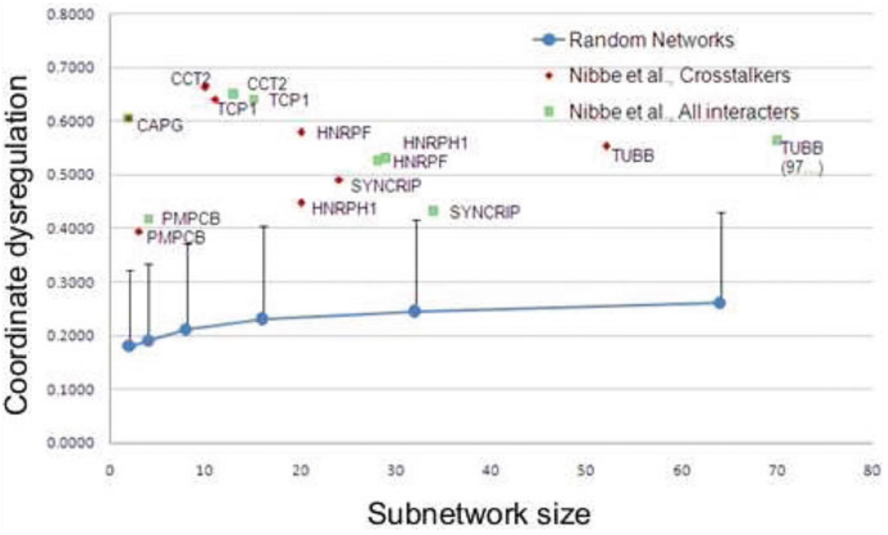


Fig. 7.13 Relationship between crosstalk to proteomic targets and coordinate mRNA-level dysregulation. *Red diamonds* represent subnetworks composed of interacting partners of a proteomic seed with significant crosstalk to all proteomic seeds. *Green squares* represent subnetworks composed of all interacting partners of a proteomic seed. The *blue curve* represents expected coordinate dysregulation for random subnetworks of given size and bars indicate one standard deviation above mean for this null distribution

Construction of interactor and crosstalker subnetworks is illustrated in Fig. 7.12.

The additive coordinate dysregulation of candidate subnetworks in human colorectal cancer is shown in Fig. 7.13. As seen in the figure, ten unique interactor subnetworks exhibit significant coordinate. For five of these subnetworks (CCT2,

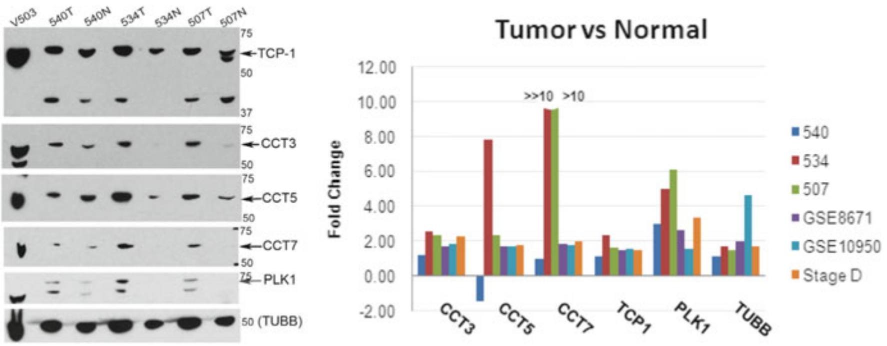


Fig. 7.14 Validation of select targets predicted to be post-transcriptionally dysregulated in TCP1 subnetwork. Immunoblot data are obtained from three (540, 534, 507) late-stage matched (N = normal/T = tumor) patient tissue biopsies not used in the original proteomic screen by Nibbe et al. [55]. Values are in kilodalton (kDa). GSE8671 and GSE10950 represent the ratio of the mean mRNA value (tumor/normal) from the respective microarray. Fold change is determined by densitometry

TCP1, SYNCRIP, HNRPF, and HNRPH1) the crosstalker version of the subnetworks is found to have enhanced mRNA-level coordinate dysregulation. These results demonstrate that significant functional association with proteomic targets can indeed be an indicator of coordinated dysregulation at the level of mRNA expression [54].

4.3 *Post-Transcriptional Dysregulation of TCP1 Subnetwork in Colorectal Cancer*

Nibbe et al. [54] observe that several of the subnetworks generated using two separate proteomic seed sets ([55] and [24]) contain proteins in common. In particular, certain subunits of the TCP1 complex exhibit marked crosstalk in the subnetwork induced by CCT2 with respect to proteomic targets discovered by Nibbe et al. [55], and TUBA1B with respect to proteomic targets discovered by Friedman et al. [24]. In addition, it is also shown [55] that certain subunits of this complex (CCT3, CCT5, and CCT7) are also significant for the late-stage CRC phenotype.

TCP1 (or TCPa) is a hetero-oligomeric complex comprised of two stacked ring structures, each composed of eight known subunits and plays a functional role in maintaining the CRC phenotype. Specifically, it was shown to be required for the proper biogenesis of PLK1, a kinase that has a critical role in cytokinesis. However, other than their role as subunits in the formation of the TCP complex little is known about the independent role, if any, of these subunits in CRC. Consequently, these targets present an opportunity for follow-on mechanistic studies. For this reason, Nibbe et al. [54] verify the protein expression of TCP1, CCT3, CCT5, CCT7, and

PLK1 by western blot in a separate cohort of three patient sample pairs not used in screening phase, and compared this to the average expression at the level of mRNA. These results are shown in Fig. 7.14. Consistent with the hypothesis suggested by the computational studies, the data not only indicate the coregulation at the level of mRNA and protein, but also reveal the wide variability of expression of these targets among individual patients. CCT3 and CCT7 are dramatically overexpressed in two patients (507 and 534), but less so in patient 540, which is similar to the pattern for PLK1.

5 Outlook

As demonstrated in this chapter, molecular networks are very promising for the integration of various -omic datasets to study complex diseases. Such integrative methods are likely to enhance utilization of omic data in personalized medicine applications by extracting information from multiple types omic data that is beyond the reach of a single type of omic data. In particular, researchers are increasingly suggesting that use of molecular data is likely to enhance GWAS through prediction of genetic interactions, as well as investigation of the mechanistic bases of identified genetic associations. Furthermore, availability of next generation sequencing platforms enables more reliable monitoring of gene expression, providing functional data for detailed study of these mechanisms. As these mechanisms are investigated more in detail, there will be more need for generation of proteomic, as well as high quality interactomic data. If these data and novel algorithms are supported by novel network modeling paradigms that can capture complex interactions more accurately, integrative omics is likely to provide significant insights into the network dynamics of complex diseases.

Acknowledgments This work was supported by National Science Foundation CAREER Award CCF-0953195, and National Institutes of Health Grant UL1-RR024989 from the National Center for Research Resources (Clinical and Translational Science Awards).

References

1. E. Adie, R. Adams, K. Evans, D. Porteous, and B. Pickard. SUSPECTS: enabling fast and effective prioritization of positional candidates. *Bioinformatics*, 22(6):773–774, 2006.
2. A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, John I. Powell, Liming Yang, Gerald E. Marti, Troy Moore, Jr James Hudson, Lisheng Lu, David B. Lewis, Robert Tibshirani, Gavin Sherlock, Wing C. Chan, Timothy C. Greiner, Dennis D. Weisenburger, James O. Armitage, Roger Warnke, Ronald Levy, Wyndham Wilson, Michael R. Grever, John C. Byrd, David Botstein, Patrick O. Brown, and Louis M. Staudt. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, Feb 2000.

3. Dimitris Anastassiou. Computational analysis of the synergy among multiple interacting genes. *Mol Syst Biol*, 3(83), 2007.
4. G. Bebek and J. Yang. PathFinder: Mining signal transduction pathway segments from protein protein interaction networks. *BMC Bioinformatics*, 8(335), 2007.
5. M. Beckerman. *Molecular and Cellular Signaling*. Springer, 2006.
6. D. G. Beer, S. L. Kardia, C. C. Huang, T. J. Giordano, A. M. Levin, D. E. Misek, L. Lin, G. Chen, T. G. Gharib, D. G. Thomas Michelle, L. Lizyness, Rork Kuick, Satoru Hayasaka, Jeremy M.G. Taylor, Mark D. Iannettoni, Mark B. Orringer, and Samir Hanash. Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nat. Med.*, 8:816–824, 2002.
7. Annamaria Bevilacqua, Maria Cristina Ceriani, Sergio Capaccioli, and Angelo Nicolin. Post-transcriptional regulation of gene expression by degradation of messenger RNAs. *Journal of Cellular Physiology*, 195(3):356–372, 2003.
8. Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, volume 30, pages 107–117, 1998.
9. Han G. Brunner and Marc A. van Driel. From syndrome families to functional genomics. *Nat Rev Genet*, 5(7):545–551, 2004.
10. Lawrence Cabusora, Electra Sutton, Andy Fulmer, and Christian V. Forst. Differential network expression during drug and stress response. *Bioinformatics*, 21(12):2898–2905, 2005.
11. J. F. Cáceres and A. R. Kornblihtt. Alternative splicing: multiple control mechanisms and involvement in human disease. *Trends Genet*, 18(4):186–193, April 2002.
12. J. Chang, M. R. Chance, C. Nicholas, N. Ahmed, S. Guilmeau, M. Flandez, D. S. Wyun D. Wang, S. Nasser, and J. M. Albanese. Proteomic changes during intestinal cell maturation in vivo. *J Proteomics*, 71(5):530–546, 2008.
13. Jing Chen, Bruce Aronow, and Anil Jegga. Disease candidate gene identification and prioritization using protein interaction networks. *BMC Bioinformatics*, 10(1):73, 2009.
14. Jing Chen, Eric E. Bardes, Bruce J. Aronow, and Anil G. Jegga. Toppgene suite for gene list enrichment analysis and candidate gene prioritization. *Nucleic acids research*, 37(Web Server issue):gkp427+, July 2009.
15. Salim A. Chowdhury and Mehmet Koyutürk. Identification of coordinately dysregulated subnetworks in complex phenotypes. In *Pacific Symposium on Biocomputing*, pages 133–144, 2010.
16. Salim A. Chowdhury, Rod K. Nibbe, Mark R. Chance, and Mehmet Koyutürk. Subnetwork state functions define dysregulated subnetworks in cancer. In *14th Intl Conf. Research in Computational Molecular Biology (RECOMB10)*, pages 80–95, 2010.
17. Hon Nian Chua, Wing-Kin Sung, and Limsoon Wong. Exploiting indirect neighbours and topological weight to predict protein function from protein–protein interactions. *Bioinformatics*, 22(13):1623–1630, 2006.
18. Han-Yu Chuang, Eunjung Lee, Yu-Tsueng Liu, Doheon Lee, and Trey Ideker. Network-based classification of breast cancer metastasis. *Mol Syst Biol*, 3, October 2007.
19. Şerban Nacu, Rebecca Critchley-Thorne, Peter Lee, and Susan Holmes. Gene expression network analysis and applications to immunology. *Bioinformatics*, 23(7):850–858, 2007.
20. Lyris M. F. de Godoy, Jesper V. Olsen, Jürgen Cox, Michael L. Nielsen, Nina C. Hubner, Florian Fröhlich, Tobias C. Walther, and Matthias Mann. Comprehensive mass-spectrometry-based proteome quantification of haploid versus diploid yeast. *Nature*, 2008.
21. Sinan Erten and Mehmet Koyutürk. Role of centrality in network-based prioritization of disease genes. In *Proceedings of the 8th European Conf. Evolutionary Computation, Machine Learning, and Data Mining in Bioinformatics (EVOBIO'10)*, volume LNCS 6023, pages 13–25, 2010.
22. Rob M. Ewing, Peter Chu, Fred Elisma, Hongyan Li, Paul Taylor, Shane Climie, Linda McBroom-Cerajewski, Mark D. Robinson, Liam O'Connor, Michael Li, Rod Taylor, Moyez Dharsee, Yuen Ho, Adrian Heilbut, Lynda Moore, Shudong Zhang, Olga Ornatsky, Yury V. Bukhman, Martin Ethier, Yinglun Sheng, Julian Vasilescu, Mohamed Abu-Farha, Jean-Philippe P. Lambert, Henry S. Duewel, Ian I. Stewart, Bonnie Kuehl, Kelly Hogue, Karen

Colwill, Katharine Gladwish, Brenda Muskat, Robert Kinach, Sally-Lin L. Adams, Michael F. Moran, Gregg B. Morin, Thodoros Topaloglou, and Daniel Figeys. Large-scale mapping of human protein-protein interactions by mass spectrometry. *Molecular systems biology*, 3, 2007.

23. L. Franke, H. Bakel, L. Fokkens, E. D. de Jong, M. Egmont-Petersen, and C. Wijmenga. Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *Am J Hum Genet*, 78(6):1011–1025, June 2006.
24. D. B. Friedman, S. Hill, J. W. Keller, N. B. Merchant, S. E. Levy, R. J. Coffey, and R. M. Caprioli. Proteome analysis of human colon cancer by two-dimensional difference gel electrophoresis and mass spectrometry. *Proteomics*, 4(3):793–811, March 2004.
25. S. Ghaemmaghami, W. K. Huh, K. Bower, R. W. Howson, A. Belle, N. Dephoure, E. K. O'Shea, and J. S. Weissman. Global analysis of protein expression in yeast. *Nature*, 425(6959):737–741, October 2003.
26. Anne M. Glazier, Joseph H. Nadeau, and Timothy J. Aitman. Finding Genes That Underlie Complex Traits. *Science*, 298(5602):2345–2349, 2002.
27. Kwang-Il Goh, Michael E. Cusick, David Valle, Barton Childs, Marc Vidal, and Albert-LÃ¡szlÃ Barabási. The human disease network. *PNAS*, 104(21):8685–8690, 2007.
28. T. R. Golub, D. K. Slonim, P. Tamayo, M. Gaasenbeek C. Huard, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, Oct 1999.
29. T. J. Griffin, S. P. Gygi, T. Ideker, B. Rist, J. Eng, L. Hood, and R. Aebersold. Complementary profiling of gene expression at the transcriptome and proteome levels in saccharomyces cerevisiae. *Molecular & Cellular Proteomics*, 1(4):323–333, April 2002.
30. Zheng Guo, Yongjin Li, Xue Gong, Chen Yao, Wencai Ma, Dong Wang, Yanhui Li, Jing Zhu, Min Zhang, Da Yang, and Jing Wang. Edge-based scoring and searching method for identifying condition-responsive protein–protein interaction sub-network. *Bioinformatics*, 23(16): 2121–2128, 2007.
31. Daniel Hanisch, Alexander Zien, Ralf Zimmer, and Thomas Lengauer. Co-clustering of biological networks and gene expression data. In *ISMB*, pages 145–154, 2002.
32. G. J. Hannon. Rna interference. *Nature*, 418(6894):244–251, July 2002.
33. Vassily Hatzimanikatis and Kelvin H. Lee. Dynamical analysis of gene networks requires both mRNA and protein expression information. *Metabolic Engineering*, 1(4):275–281, 1999. 3-6 FIELD Section Title:Biochemical Genetics Cargill Bioscience Division, Wayzata, MN, USA. FIELD URL: written in English.
34. Joel N. Hirschhorn and Mark J. Daly. Genome-wide association studies for common diseases and complex traits. *Nature Reviews Genetics*, 6(2):95–108, February 2005.
35. T. Ideker and R. Sharan. Protein networks in disease. *Genome research*, 18(4):644–652, 2008.
36. Trey Ideker, Owen Ozier, Benno Schwikowski, and Andrew F. Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. In *ISMB*, pages 233–240, 2002.
37. Shaul Karni, Hermona Soreq, and Roded Sharan. A network-based method for predicting disease-causing genes. *Journal of Computational Biology*, 16(2):181–189, 2009.
38. B. P. Kelley, R. Sharan, R. M. Karp, T. Sittler, D. E. Root, B. R. Stockwell, and T. Ideker. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *PNAS*, 100(20):11394–11399, 2003.
39. Ryan Kelley and Trey Ideker. Systematic interpretation of genetic interactions using protein networks. *Nature Biotechnology*, 23(5):561–566, May 2005.
40. Yoo-Ah Kim, Stefan Wuchty, and Teresa M. Przytycka. Simultaneous identification of causal genes and dys-regulated pathways in complex diseases. In *Proc. 14th Int'l Conf. Research in Computational Molecular Biology (RECOMB'10)*, volume LNCS 6044, pages 263–280, 2010.
41. Sebastian Köhler, Sebastian Bauer, Denise Horn, and Peter N. Robinson. Walking the interactome for prioritization of candidate disease genes. *Am J Hum Genet*, 82(4):949–958, 2008.
42. M. Koyutürk, Y. Kim, U. Topkara, S. Subramaniam, W. Szpankowski, and A. Grama. Pairwise alignment of protein interaction networks. *J Comput Biol*, 13(2):182–199, 2006.

43. Kasper Lage, E. Karlberg, Zenia Storling, Pall Olason, Anders Pedersen, Olga Rigina, Anders Hinsby, Zeynep Tumer, Flemming Pociot, Niels Tommerup, Yves Moreau, and Soren Brunak. A human phenome-interactome network of protein complexes implicated in genetic disorders. *Nat Bio.*, 25(3):309–316, 2007.
44. Jacques Lapointe, Chunde Li, John P. Higgins, Matt van de Rijn, Eric Bair, Kelli Montgomery, Michelle Ferrari, Lars Egevad, Walter Rayford, Ulf Bergerheim, Peter Ekman, Angelo M. DeMarzo, Robert Tibshirani, David Botstein, Patrick O. Brown, James D. Brooks, and Jonathan R. Pollack. Gene expression profiling identifies clinically relevant subtypes of prostate cancer. *PNAS*, 101:811–816, Jan 2004.
45. J. Li, B. Lee, and A. S. Lee. Endoplasmic reticulum stress-induced apoptosis. *The Journal of Biological Chemistry*, 281(11):7260–7270, 2006.
46. Manway Liu, Arthur Liberzon, Sek W. Kong, Weil R. Lai, Peter J. Park, Isaac S. Kohane, and Simon Kasif. Network-based analysis of affected biological processes in type 2 diabetes models. *PLoS Genetics*, 3(6):e96+, June 2007.
47. L Lovász. Random walks on graphs: A survey. *Combinatorics, Paul Erdos is Eighty*, 2:353–398, 1996.
48. Kathy Macropol, Tolga Can, and Ambuj Singh. Rrw: repeated random walks on genome-scale protein networks for local cluster discovery. *BMC Bioinformatics*, 10(1):283, 2009.
49. Hiroshi Mamitsuka, Yasushi Okuno, and Atsuko Yamaguchi. Mining biologically active patterns in metabolic pathways using microarray expression profiles. *SIGKDD Explor. Newsl.*, 5(2):113–121, 2003.
50. SD Markowitz and MM Bertagnolli. Molecular origins of cancer: Molecular basis of colorectal cancer. *N Engl J Med*, 361(25):2449–2460, Dec 2009.
51. F. A. Middleton, K. Mirnics, J. N. Pierri, D. A Lewis, and P. Levitt. Gene expression profiling revelas alterations of specific metabolic pathways in schizophrenia. *The Journal of Neuroscience*, 22(7):2718–2729, 2002.
52. T. M. Murali and Corban G. Rivera. Network legos: Building blocks of cellular wiring diagrams. In *RECOMB*, pages 47–61, 2007.
53. Elena Nabieva, Kam Jim, Amit Agarwal, Bernard Chazelle, and Mona Singh. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinf.*, 21:i302–310, 2005.
54. R. Nibbe, M. Koyutürk, and M. Chance. Integrative proteomics approaches to identify important sub-networks in human colorectal cancer. *PLoS Computational Biology*, 6(1):e1000639, 2010.
55. R. K. Nibbe, R. Ewing, L. Myeroff, M. Markowitz, and M. Chance. Discovery and scoring of protein interaction sub-networks discriminative of late stage human colon cancer. *Mol Cell Prot*, 9(4):827–845, 2009.
56. Alexandra C. Nica and Emmanouil T. Dermitzakis. Using gene expression to investigate the genetic basis of complex disorders. *Human molecular genetics*, 17(R2):ddn285–134, October 2008.
57. Martin Oti, Berend Snel, Martijn A Huynen, and Han G Brunner. Predicting disease genes using protein-protein interactions. *J Med Genet*, page jmg.2006.041376, 2006.
58. Jayesh Pandey, Mehmet Koyutürk, and Ananth Grama. Functional characterization and topological modularity of molecular interaction networks. *BMC Bioinformatics*, 11(Suppl. 1):S35, 2010.
59. Jayesh Pandey, Mehmet Koyutürk, Shankar Subramaniam, and Ananth Grama. Functional coherence in domain interaction networks. *Bioinformatics*, 24(16):i28–34, 2008.
60. Vishal N. Patel, Gurkan Bebek, John M. Mariadason, Donghai Wang, Leonard H. Augenlicht, and Mark R. Chance. Prediction and testing of biological networks underlying intestinal cancer. *PLoS ONE*, 5(9):e12497, 09 2010.
61. K. R. Patil and J. Nielsen. Uncovering transcriptional regulation of metabolism by using metabolic network topology. *Proc Natl Acad Sci U S A*, 102(8):2685–2689, February 2005.
62. C. M. Perou, T. Srlie, M. B. Eisen, M. van de Rijn, S. S. Jeffrey, C. A. Rees, J. R. Pollack, D. T. Ross, H. Johnsen, L. A. Akslen, ystein Fluge, Alexander Pergamenschikov, Cheryl Williams,

Shirley X. Zhu, Per E. Lnning, Anne-Lise Brresen-Dale, Patrick O. Brown, and David Botstein. Molecular portraits of human breast tumours. *Nature*, 406:747–752, Aug 2000.

63. Dilip Rajagopalan and Pankaj Agarwal. Inferring pathways from gene lists using a literature-derived network of biological relationships. *Bioinformatics*, 21(6):788–793, 2005.
64. D. R. Rhodes and A. M. Chinnaiyan. Integrative analysis of the cancer transcriptome. *Nat Genet*, 37 Suppl, June 2005.
65. Karen Sachs, Omar Perez, Dana Pe'er, Douglas A. Lauffenburger, and Garry P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, April 2005.
66. M. S. Scott, T. Perkins, S. Bunnell, F. Pepin, D. Y. Thomas, and M. Hallett. Identifying regulatory subnetworks for a set of genes. *Molecular & Cellular Proteomics*, pages 683–692, 2005.
67. Eran Segal, Haidong Wang, and Daphne Koller. Discovering molecular pathways from protein interaction and gene expression data. In *ISMB (Supplement of Bioinformatics)*, pages 264–272, 2003.
68. Tomer Shlomi, Moran N N. Cabili, Markus J J. Herrgård, Bernhard Ø O. Palsson, and Eytan Ruppin. Network-based prediction of human tissue-specific metabolism. *Nature biotechnology*, August 2008.
69. D. K. Slonim. From patterns to pathways: gene expression data analysis comes of age. *Nat Genet*, 32 Suppl:502–508, December 2002.
70. Raoul Tibes, YiHua Qiu, Yiling Lu, Bryan Hennessy, Michael Andreeff, Gordon B. Mills, and Steven M. Kornblau. Reverse phase protein array: validation of a novel proteomic technology and utility for analysis of primary leukemia specimens and hematopoietic stem cells. *Mol Cancer Ther.*, 5:2512–2521, 2006.
71. Hanghang Tong and Christos Faloutsos. Center-piece subgraphs: problem definition and fast solutions. In *KDD '06: Proceedings of the 12th ACM SIGKDD*, pages 404–413, NY, USA, 2006. ACM.
72. Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Random walk with restart: fast solutions and applications. *Knowledge and Information Systems*, 14(3):327–346, 2008.
73. S. Tornow and H. W. Mewes. Functional modules by relating protein interaction networks and gene expression. *Nucleic Acids Research*, 31(21):6283–6289, 2003.
74. Frances Turner, Daniel Clutterbuck, and Colin Semple. Pocus: mining genomic sequence annotation to predict disease genes. *Genome Biology*, 4(11):R75, 2003.
75. Igor Ulitsky, Richard M. Karp, and Ron Shamir. Detecting disease-specific dysregulated pathways via analysis of clinical expression profiles. In *RECOMB*, pages 347–359, 2008.
76. Fabio Vandin, Eli Upfal, and Benjamin J. Raphael. Algorithms for detecting significantly mutated pathways in cancer. In *Proc. 14th Int'l Conf. Research in Computational Molecular Biology (RECOMB'10)*, volume LNCS 6044, pages 506–521, 2010.
77. Oron Vanunu, Oded Magger, Eytan Ruppin, Tomer Shlomi, and Roded Sharan. Associating genes and protein complexes with disease via network propagation. *PLoS computational biology*, 6(1):e1000641+, January 2010.
78. Jean-Philippe Vert and Minoru Kanehisa. Extracting active pathways from gene expression data. In *ECCB*, pages 238–244, 2003.
79. Ramana Vishnubhotla, Shan Sun, Jameela Huq, Marinka Bulic, and Anil Ramesh. Rock-ii mediates colon cancer invasion via regulation of mmp-2 and mmp-13 at the site of invadopodia as revealed by multiphoton imaging. *Laboratory Investigation*, 87:1149–1158, 2007.
80. Zhong Wang, Mark Gerstein, and Michael Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, January 2009.
81. John Watkinson, Xiaodong Wang, Tian Zheng, and Dimitris Anastassiou. Identification of gene interactions associated with disease from gene expression data using synergy networks. *BMC Systems Biology*, 2(1), 2008.
82. E. Yohannes, J. Chang, G. J. Christ, K. P. Davies, and M. R. Chance. Proteomics analysis identifies molecular targets related to diabetes mellitus-associated bladder dysfunction. *Mol Cell Proteomics*, 7(7):1270–1285, 2008.

Chapter 8
Moving Toward Genome-Scale Kinetic Models: The Mass Action Stoichiometric Simulation Approach

Aarash Bordbar and Bernhard Ø. Palsson

Abstract Kinetic models are used to describe cellular metabolism. Traditional models are based on enzymatic information obtained from in vitro experiments. In vitro data is inaccurate for in vivo modeling and is difficult to scale to large metabolic networks. Due to the impeding availability of metabolomic and fluxomic data types, we present an alternative kinetic modeling approach. Mass action stoichiometric simulation (MASS) models are scalable kinetic models that detail in vivo metabolic transformations. MASS formulation is a "middle-out" approach involving the use of a genome-scale metabolic network as a scaffold to map fluxomic and metabolomic measurements. Multiple binding states of enzymes can be explicitly added to account for regulatory effects. There are practical challenges with data completeness and quality of MASS models, but they do represent scalable kinetic models that exhibit biological properties such as time scale decomposition and account for regulation.

1 Introduction

Metabolism is a universal and complex biochemical process that provides the energy and material resources for living organisms. Since the first whole genome-sequences appeared in the mid 1990s, there has been interest in reconstructing large-scale metabolic networks. Prior to this time, large-scale metabolic network reconstruction existed as mosaics of metabolic capabilities found in multiple organisms. With the availability of genome sequences for target organisms, such maps graduated to genome-scale reconstructions that are organism specific. This process started

B.Ø. Palsson (✉)
Department of Bioengineering, University of California – San Diego,
9500 Gilman Dr., La Jolla, CA, USA
e-mail: palsson@ucsd.edu

M. Koyutürk et al. (eds.), *Functional Coherence of Molecular Networks in Bioinformatics*, DOI 10.1007/978-1-4614-0320-3_8,
© Springer Science+Business Media, LLC 2012

with *Haemophilus influenza* [4]. Genome-scale metabolic reconstructions play an integral part in systems biology and have proven to play an important part in addressing biologically relevant problems [8, 18].

Due to the immense size and lack of kinetic parameter data for genome-scale reconstructions, constraint-based reconstruction and analysis (COBRA) methods have been mainly employed to characterize these networks [1]. COBRA methods assume a steady state and utilize linear programming to analyze the biochemical transformations in metabolism. Although this approach can be applied to bacterial reconstructions under certain circumstances [9], biological processes are inherently dynamic, especially in higher-order organisms.

Traditional biophysical approaches to model building have been limited to models with a few dozen variables detailing specific enzymes or pathways [12, 16, 17]. This approach does not scale easily. In this chapter, we present a new, data-driven approach to building large-scale kinetic models. The mass action stoichiometric simulation (MASS) approach utilizes metabolic network reconstructions as a stoichiometric texture against which high-throughput (omics) data is used to determine condition-specific pseudo-elementary rate constants (PERCs). We outline the method to build these models, describe the challenges of the modeling approach, provide a whole red blood cell model, and outline the future of the MASS modeling approach.

1.1 Traditional Kinetic Modeling

Traditional kinetic modeling of metabolism involves analytically solving enzyme rate equations, making key assumptions, and parameterizing the resulting rate law using in vitro enzyme data. To characterize a metabolic pathway with multiple enzymes, each reaction's rate law is modularly solved and then combined. The traditional modeling approach has difficulty in describing in vivo metabolism due to the use of in vitro enzyme data and the lack of scalability.

MASS models are an alternative approach that uses emerging omics data sets to determine kinetic parameters. The process uses in vivo data, is scalable, and can explicitly represent regulatory interactions. The advantages and disadvantages of traditional kinetic and MASS modeling is outlined in Table 8.1.

1.2 Entering the Omics World

We are entering a new-age of biological data. Omics is an emerging research paradigm to generate large amounts of high-throughput data from a systems perspective. The ability to generate these large, biological data sets has helped push the fields of bioinformatics and systems biology [23]. Terms like genomics, proteomics, and transcriptomics are becoming common place in biology and other related fields.

Table 8.1 Comparison of the advantages and disadvantages of the MASS and traditional approach of enzyme kinetic modeling. Adapted from [15]

	MASS models	Traditional kinetic models
Model building	Omics driven algorithmic approach	Modularized enzyme approach
Protein activity resolution	Explicit stoichiometric definition, allows direct integration of omics data from a networks perspective	Requires quasi steady-state and quasi-equilibrium assumptions to account for inability to describe enzyme intermediates and protein–protein interactions
Scalability	Good, though must deal with more variables	Difficult due to the case-by-case treatment of additional enzymes
Condition specificity	Dependent on steady state due to the lumping of many factors into PERCs	Accounts for many details, biophysical in nature making the models global
Data quality	In vivo data used, dependent on quality of concentration and fluxes measurements	Utilizes in vitro data
Data completeness	Limited by coverage of emerging omic data sets	Limited by inability to characterize all enzymes and measure all metabolites in vivo

MASS models attempt to utilize the information flowing out of this budding field to build large-scale kinetic models. MASS models use three main types of omics data: genomics, metabolomics, and fluxomics. In addition, transcriptomics and proteomics are becoming important data sets to tailor MASS models. Throughout the rest of this chapter, we will point out the use of omics data sets.

2 Building and Analyzing MASS Models

In this section, we provide a workflow for building and analyzing MASS models. MASS model building involves a “middle-out” approach that combines a bottom-up reconstructed metabolic network as a scaffold to map top-down data measurements (Fig. 8.1). Using omics data, large-scale kinetic models can be properly parameterized, simulated, and analyzed (Fig. 8.3).

2.1 Data Sources and Requirements

Before describing the dynamic properties, the network must first be established. Whole genome-sequencing allows reconstructing genome-scale metabolic networks. For a description for building high-quality genome-scale metabolic

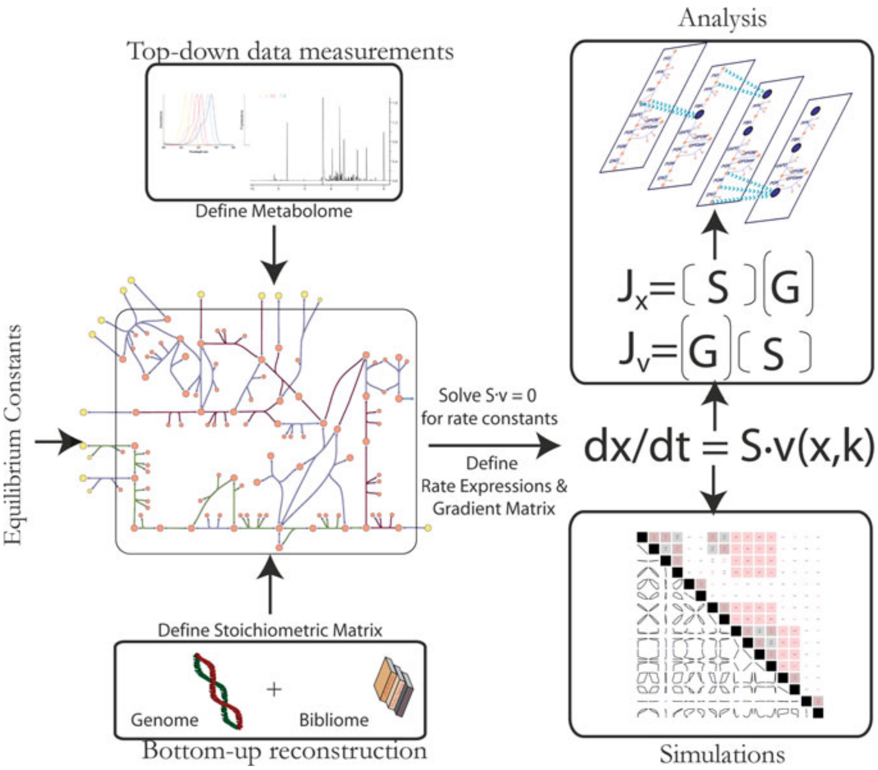


Fig. 8.1 An overview of MASS models construction and analysis. The MASS approach involves a “middle-out” approach of combining a bottom-up reconstruction as a scaffold with top-down data measurements to build a kinetic model. Once the dynamic mass balances have been defined, dynamic simulations and dynamic analysis of the Jacobian help define the dynamic properties of the network. Adapted from [15]

reconstructions, we refer you to the following protocol [21]. A stoichiometric matrix (**S**) can be generated from the reconstruction. The rows of the stoichiometric matrix represent the metabolites or “nodes” of the network, while the columns represent the transformations or “links.” The stoichiometric matrix is used as a scaffold for mapping data measurements. Three major data measurements are required to properly characterize a dynamic MASS model: (1) steady-state metabolite concentrations ($\mathbf{x}_{\mathbf{ss}}$), (2) steady-state reaction fluxes ($\mathbf{v}_{\mathbf{ss}}$), and (3) equilibrium constants ($\mathbf{K}_{\mathbf{eq}}$). Metabolite concentrations and reaction fluxes can be obtained from metabolomic and fluxomic measurements, respectively. Equilibrium constants are determined through literature search. The stoichiometric matrix derived from the genome-scale metabolic reconstruction and the data measurements serve as the building blocks of MASS models (Fig. 8.2).

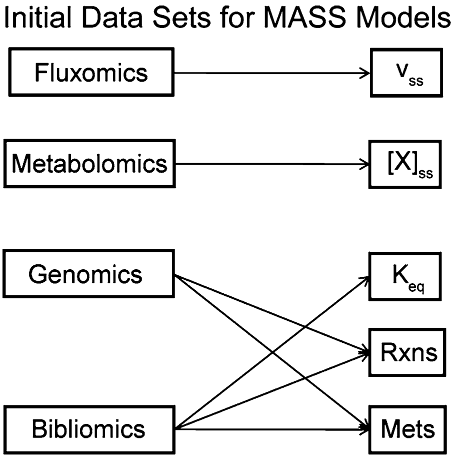


Fig. 8.2 The MASS approach requires four omics data types for construction. Available genome sequences combined with primary literature can help define reactions and metabolites to build a genome-scale reconstruction and stoichiometric matrix. Fluxomics and metabolomics provide steady-state fluxes and metabolite concentrations, respectively

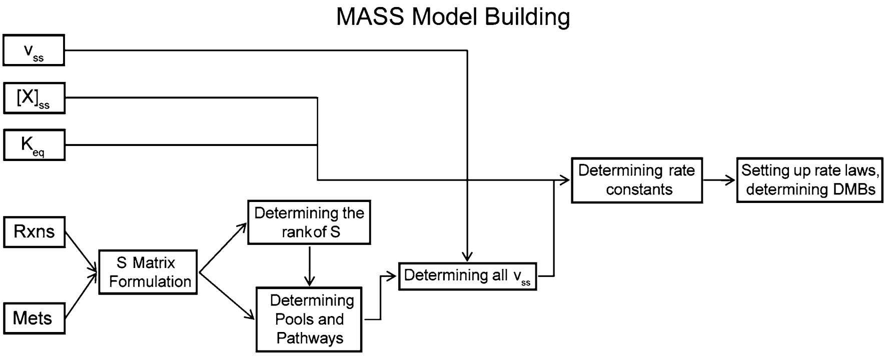


Fig. 8.3 Workflow of building MASS models. First, a stoichiometric matrix must be constructed or adapted from an existing constraint-based genome-scale reconstruction. The null-space is used to define $\mathbf{n} - \mathbf{r}$ pathways for the system. Fluxomic data is used to weight the pathways and lay a steady-state flux distribution on the network. Alongside the steady-state metabolite concentrations and equilibrium constants, the pseudo-elementary rate constants are defined. At this point, the model is fully parameterized. Ensuing rate laws and dynamic mass balances are calculated

2.2 Defining the Null Spaces and Setting the Steady State

We now present the workflow for building MASS models (Fig. 8.3). After building the stoichiometric matrix and obtaining the proper omics data, the null spaces are defined. The left null space contains information on the time-invariant concentration (or “hard”) pools. We will discuss “hard” and “soft” pools later in this chapter. The right null space details the basis vectors that make up the steady-state pathways, or extreme pathways [19] of the network. The stoichiometric matrix has dimensions $\mathbf{m} \times \mathbf{n}$. Defining $\mathbf{n} - \mathbf{r}$ pathways with known flux rates, where $\mathbf{r}$ is the rank of the matrix, allows calculation of a unique flux steady state of the entire network ($\mathbf{v_{ss}}$). This is done by left multiplying the extreme pathways ($\mathbf{N_{expa}}$) by the fluxomic measurements ($\mathbf{v_{flux}}$) (8.1).

$$(\mathbf{v_{flux}} \cdot \mathbf{N_{expa}})^T = \mathbf{v_{ss}} \tag{8.1}$$

The second component of the steady state is the metabolite concentrations. With emerging metabolomic technologies and data, steady-state metabolite concentrations can be mapped to the system.

2.3 Determining the Pseudo-Elementary Rate Constants

The steady-state concentrations, fluxes, and equilibrium constants provide the necessary parameters to solve for the PERCs. The kinetic rate constants are called “pseudo-elementary” because they are condition-dependent to the defined steady state. Simple algebra solves for the PERCs. A prototypical example is shown below to illustrate:

$$A + B \Leftrightarrow C \tag{8.2}$$

$$\mathbf{v_{ss}} = k_+ \left([A]_{\mathbf{ss}}[B]_{\mathbf{ss}} - \frac{[C]_{\mathbf{ss}}}{K_{\mathbf{eq}}} \right) \tag{8.3}$$

$$k_+ = \frac{\mathbf{v_{ss}}}{\left([A]_{\mathbf{ss}}[B]_{\mathbf{ss}} - \frac{[C]_{\mathbf{ss}}}{K_{\mathbf{eq}}} \right)} \tag{8.4}$$

This process is repeated for all reactions in the stoichiometric matrix. Now, all the necessary parameters are defined to write the rate laws for the reactions and the dynamic mass balances for the metabolites. The dynamic mass balances are generated automatically by using:

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{S} \cdot \mathbf{v}, \tag{8.5}$$

where $\mathbf{v}$ is a vector containing all fluxes pertaining to the stoichiometric matrix, $\mathbf{S}$. The dynamic mass balances, stoichiometric matrix, and steady-state data sets combine to form the MASS model.

2.4 Aggregating Variables into Pools

With the proper definition of the system, several different analytical methods can be used to characterize and probe the network dynamics of MASS models (Fig. 8.4). We begin by describing the aggregation of concentrations into pools. Pools can be

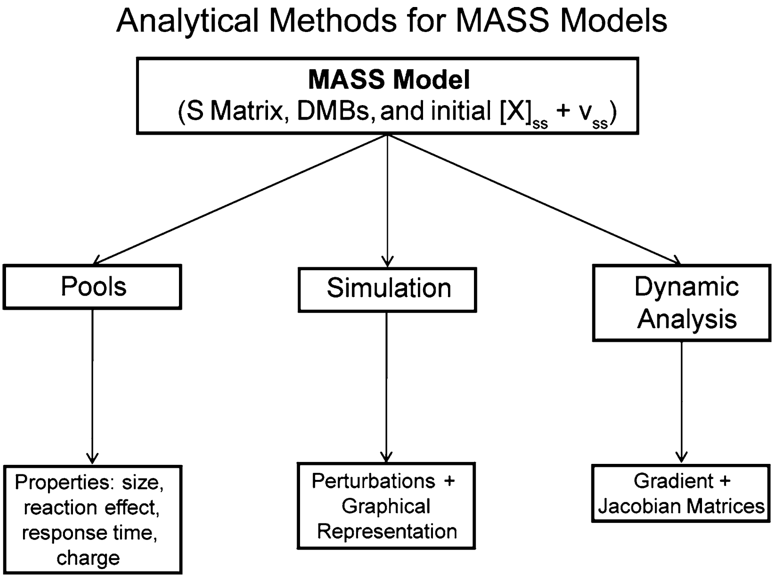

Analytical Methods for MASS Models
MASS Model
(S Matrix, DMBs, and initial $[X]_{ss}$ + v_{ss})
Pools
Simulation
Dynamic Analysis
Properties: size, reaction effect, response time, charge
Perturbations + Graphical Representation
Gradient + Jacobian Matrices

Fig. 8.4 Three analytical methods are used for characterizing the dynamic network properties of MASS models. Pools are aggregates of metabolite concentrations which help temporally decompose the network. Dynamic simulation is used to determine how the system responds to perturbations. Dynamic analysis involves utilizing the gradient and Jacobian matrices to temporally decompose the network and mathematically form aggregate variables

formed based on time scale hierarchy, chemical characteristics, and conservation. First, biological systems are characterized by chemical transformations on multiple time scales. The separation of time constants of reactions leads to the aggregation of concentrations in a hierarchical fashion that can be physiologically significant. Such pools are called "soft" and are time-variant and are usually formed through intuition. However, "soft" pools can also be determined mathematically by temporal decomposition of the Jacobian matrix [14], which will be discussed later. Second, "soft" pools can be constructed by chemical characteristics. Such pools include quantities of total inventory, such as high-energy phosphate bonds or redox equivalents. The third type of pool is time-invariant and called "hard." "Hard" pools are conserved quantities in the MASS model and are determined by linear combinations of the basis vectors of the left null space.

Pools serve two main purposes. First, the ability to aggregate concentrations based on the time scale of chemical transformation is critical in understanding the temporal hierarchy of network dynamics. As discussed earlier, physiological functions occur at starkly different rates. Temporally pooling variables provides a method to determine how biological events organize themselves by reducing the dimensionality of the biochemical network.

Second, pools give insight to the metabolic state of the system. For example, a typical pool used in MASS models is for the occupancy of high-energy phosphate bonds: 2ATP + ADP. A second pool is defined for the capacity of high-energy phosphate bonds: 2(ATP + ADP + AMP). The ratio of the occupancy and capacity provides the adenylate energy charge of the system, a physiologically important quantity that provides insight on the state of the system. Other ratios, such as detailing the redox potential of the system, can also be enumerated.

2.5 Dynamic Simulation

MASS models are composed of ordinary differential equations that constitute the dynamic mass balances. The dynamic mass balances are simulated using numeric solvers. Perturbations are done to the MASS model to investigate dynamic changes and adaptations. The simulations provide an opportunity to characterize the dynamic relationships between components within a network. Depletion of a cell's ATP and NADH reserves are common perturbations.

Ordinary differential equations are straightforward to solve using numeric solvers. A potential difficulty arises with numerical simulation due to the disparate magnitudes of time constants. The disparity makes the ordinary differential equations stiff and potentially difficult to simulate. However, current commercial ordinary differential equation solvers are adequate dealing with the numerical intractability associated with disparate time constants. As MASS models approach genome-scale and account for faster dynamics, such as enzymatic regulation and phosphorylation, new ordinary differential equation solvers and analytical methods are required to properly characterize MASS models.

Dynamic simulations are interpreted using appropriate graphical methods. Time profiles and phase portraits are plotted for metabolic concentrations, reaction fluxes, aggregate variables, and pool ratios. Time profiles provide the status of the biochemical system throughout the simulation time range. Following the ratios of pools through this time course provides physiological insight of the biological network. Phase portraits relate the interactions of concentrations, fluxes, and pools on a network scale and show a deeper temporal relationship of cause and effect.

2.6 Dynamic Analysis

Simulation tools are a powerful method for analyzing kinetic models. However, dynamic simulations are not always scalable. Dealing with high numbers of metabolites and reactions, and incorporating enzymatic regulation makes the ordinary differential equations stiff. Numerical solution of such large and stiff networks becomes computationally exhaustive. Analytical methods for dynamic properties of networks are used. Unlike numerical solution results that are condition-specific to the steady-state and chosen initial conditions, dynamic analysis methods can be generalized.

The Jacobian matrix (**J**) can be defined for the concentrations (8.7) and fluxes (8.8) of biochemical networks through the left and right multiplication of the gradient matrix (**G**) by the stoichiometric matrix, respectively. The gradient matrix contains the kinetic properties of each individual link in the network and the elements of **G** are defined as the partial derivative of the fluxes with respect to the concentrations (8.6).

$$g_{ij} = \frac{\partial \mathbf{v}_i(\mathbf{x})}{\partial \mathbf{x}_j} \tag{8.6}$$

$$\mathbf{J}_x = \mathbf{SG} \tag{8.7}$$

$$\mathbf{J}_v = \mathbf{GS} \tag{8.8}$$

The Jacobian matrices are related to the time derivative of the concentration $(\mathbf{x}')$ and flux $(\mathbf{v}')$ deviation variables. A deviation variable is the difference between the variable and its reference point. For MASS models, the reference point is the set steady state. The network dynamics of a MASS model can be described from the viewpoint of the two variables: concentrations (8.9) and fluxes (8.10).

$$\frac{\mathrm{d}\mathbf{x}'}{\mathrm{d}t} = \mathbf{J}_x\mathbf{x}' \tag{8.9}$$

$$\frac{\mathrm{d}\mathbf{v}'}{\mathrm{d}t} = \mathbf{J}_v\mathbf{v}' \tag{8.10}$$

Equation (8.9) can be used to study the temporal hierarchy of the biochemical network. The concentration Jacobian is decomposed into (8.11), where Λ is a diagonal matrix of the eigenvalues and $\mathbf{M}^{-1}$ is the modal matrix. The eigenvalues represent the negative reciprocal of the time constants.

$$\mathbf{J}_x = \mathbf{M}\,\Lambda\,\mathbf{M}^{-1} \tag{8.11}$$

The rows of the modal matrix define aggregate variables, or pools, that are dynamically independent from other variables. The pools are ordered from the fastest timescale to the slowest in the modal matrix. The columns of the modal matrix represent each moiety in the stoichiometric matrix. When two concentrations are dynamically coupled, the two columns are correlated for the rows which they are linked. Hence, pool formation can be mathematically determined by calculating the angle between the columns of the modal matrix as:

$$\Theta = \cos(\varphi_{ij}) = \frac{\left(\mathbf{M}^{-1}\right)_i^T \cdot \left(\mathbf{M}^{-1}\right)_j}{\left|\left(\mathbf{M}^{-1}\right)_i^T\right|\left|\left(\mathbf{M}^{-1}\right)_j\right|}, \tag{8.12}$$

where φ_{ij} represents the angle between the ith and jth columns of the modal matrix and the denominator terms are magnitudes of the modal matrix. As the angle approaches zero, the columns in question begin forming a pool. Using this mathematical method for calculating dynamically aggregating variables does not require specification of the time scales of interest, unlike dynamic simulation.

Though dynamic analysis is better scaled for large networks, the Jacobian matrices can become ill-conditioned and difficult to analyze. An ill-conditioned matrix has a very large condition number. The condition number of a matrix is defined as the logarithmic ratio of the largest and smallest eigenvalues. The wide range of time scales makes many MASS models ill-conditioned. Concentrations are sometimes normalized and the PERCs are recalculated to avoid a large condition number.

2.7 Enzymes

Small metabolites are not the only molecules that can be mechanistically modeled using the MASS approach. Enzymes and their resulting complexes can be explicitly represented as nodes in the stoichiometric matrix. Enzyme subnetworks are modular in nature and can be integrated into the stoichiometric matrix (Fig. 8.5). Depending on the available experimental data, different methods are used to define the biochemical transformations and molecular concentrations. Proteomic and transcriptomic data sets can be used to determine enzyme states and can be mapped onto MASS models to add context to the network dynamics.

The typical enzyme mechanism used in MASS models involves a sequential catalytic pathway with a steady-state flux matching the unregulated chemical transformation ((8.13)–(8.17)) as well as inhibiting and activating mechanisms in equilibrium ((8.18) and (8.19)). Activated enzymes have a similar catalytic mechanism ((8.13)–(8.17)).

$$E + S_1 \leftrightarrow ES_1 \tag{8.13}$$

$$ES_1 + S_2 \leftrightarrow ES_1S_2 \tag{8.14}$$

$$ES_1S_2 \leftrightarrow EP_1P_2 \tag{8.15}$$

$$EP_1P_2 \leftrightarrow EP_1 + P_2 \tag{8.16}$$

$$EP_1 \leftrightarrow E + P_1 \tag{8.17}$$

$$E + I \leftrightarrow EI \tag{8.18}$$

$$E + A \leftrightarrow EA \tag{8.19}$$

The use of enzymes and regulation in MASS models is a developing field. Few examples exist of properly tailored enzymes, but we present a regulated red blood cell MASS model with five enzyme modules later in this chapter.

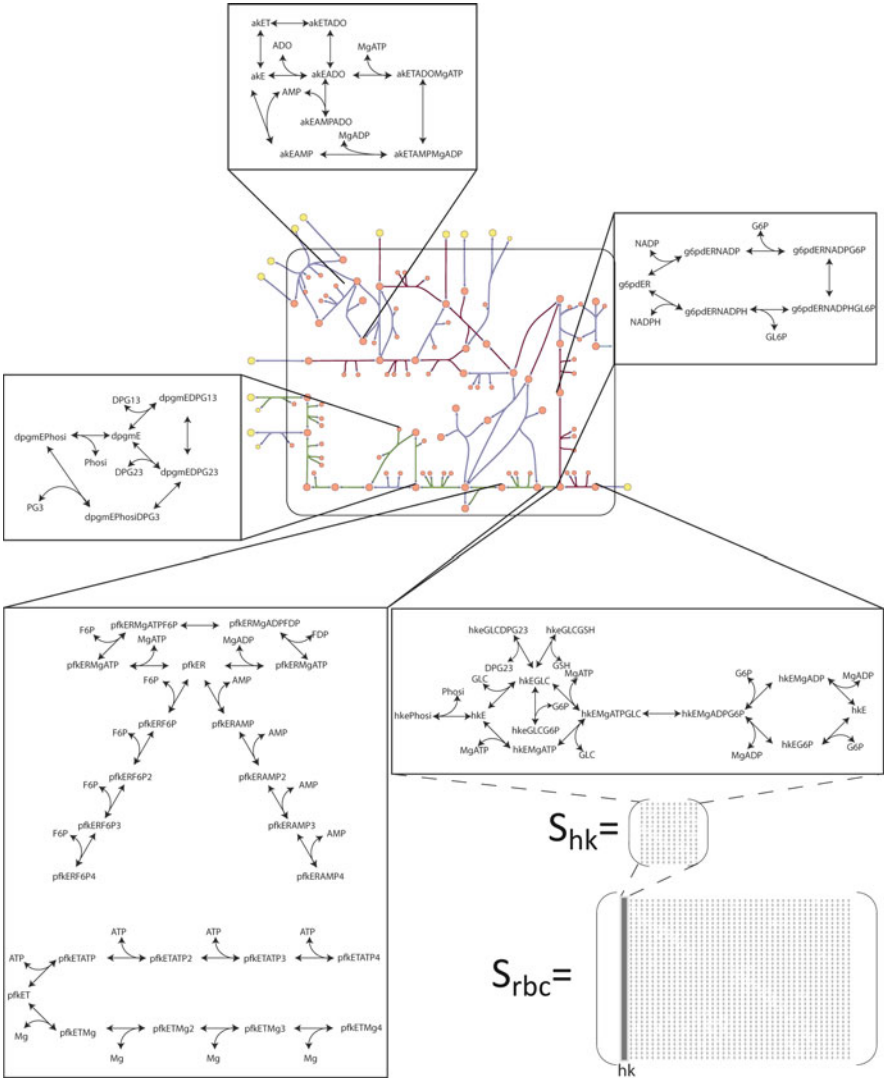


Fig. 8.5 A topological overview of the regulated red blood cell model. The reactions in the center of the figure are the metabolic transformations of the red blood cell. The five boxes surrounding this network are the enzyme modules that account for allosteric regulation. The modules are built separately and then integrated with the final stoichiometric matrix. Adapted from [15]

2.8 QC/QA Procedures

There are three main quality control/quality assurance procedures to ensure validity of the stoichiometric matrix and calculated PERCs. The first step in building MASS models is to derive a stoichiometric matrix from genomic and bibliomic data.

The procedure is mostly automated but manual curation is required to elementally balance the biochemical transformations. The elemental matrix (**E**) is represented with rows of elements and columns of metabolites. If network reactions are properly elementally balanced, right multiplying the **E** matrix by the stoichiometric matrix results in a zero matrix (8.20). When the steady-state pathways are calculated for the network, the entire pathway should have the same result when right multiplying by the **E** matrix.

$$\mathbf{E} \cdot \mathbf{S} = 0 \tag{8.20}$$

Second, numerical checks are made that the calculated PERCs are all positive. Though most reactions in cellular metabolism are from equilibrium, some reactions are close enough that if the top-down measurements have error, negative PERCs are calculated. Omics data from affected reactions are adjusted to ensure a positive kinetic rate constant.

Finally, after the rate laws have been properly parameterized, a final check is made (8.21). The steady-state metabolite concentrations are plugged into the rate laws and right multiplied with the stoichiometric matrix to result in a zero vector. The third step ensures that all parameters are specific to the steady state.

$$\mathbf{S} \cdot \mathbf{v} = 0 \tag{8.21}$$

The MASS model must meet all three criteria before proceeding to aggregating variables, dynamic analysis, and numerical simulation.

3 Challenges

Traditional kinetic models are largely inaccurate due to the lack of availability of proper data. The MASS approach is able to side-step many of the challenges faced by traditional methods. However, issues of omics data completeness and quality and scalability are a challenge for further development of MASS models. In this section, we highlight the potential challenges facing MASS models and some methods to deal with these challenges.

3.1 Data Completeness and Quality

The major challenge facing all kinetic models is data completeness. Steady-state concentrations and fluxes and equilibrium constants are the main data types required for building MASS models. It is impossible to have all the required data for every reaction for genome-scale metabolic networks.

The most readily available data is emerging to be small metabolite concentrations. Metabolomic studies of whole-cells are becoming available and will play an important role in parameterizing MASS models.

Unlike small metabolites, it is currently not possible to measure the steady-state concentrations of enzymes and their complexes under in vivo conditions. The concentrations of all applicable complex states are estimated in MASS models. The data usually available for metabolic enzymes include dissociation constants and total enzyme concentration. Using this data, proper estimation of enzyme complex concentrations is made. Enzyme concentrations in dead-end reactions, such as inhibitory steps, are assumed to be at equilibrium. Enzyme concentrations in catalytic linear pathways are analytically solved by estimating kinetic parameters. New sampling methods are being developed to address the lack of enzymatic data. The exact concentration of each enzyme complex is not quite important for MASS models. Rather the ratio of active to inactive states is more pertinent and the current estimation techniques described are adequate.

The steady-state flux state is the second top-down measurement required for MASS models. Fluxomic experiments determine flux rates through key reactions in cells. However, the flux state of all reactions in a metabolic network cannot be determined. As described in (8.1), the flux-steady state is set by specifying flux rates of only $\mathbf{n} - \mathbf{r}$ extreme pathways. Picking proper pathways that cover both the network as well as have available fluxomic data is critical to accurately setting the flux steady state. Linear programming can be used to determine the best combination of the two criteria in determining the extreme pathways used by first assigning desirability weights to the reactions.

Futile cycles in networks are known as Type III pathways. These pathways carry a zero net flux and cannot be properly characterized in MASS models. Monte Carlo sampling methods that probe the solution space of the *k*-cone have been shown to be promising for determining PERCs for these reactions [6].

There are data issues with the equilibrium constants as well. Equilibrium constants are determined by perusing the literature. This process is manageable for small networks but becomes cumbersome for networks approaching genome-scale due to the lack of available data for less characterized enzymes. It is possible to estimate the unavailable equilibrium constants but a bad estimation can result in negative PERCs. If reliable metabolomic data is available, sampling of the *k*-cone provides potential forward and reverse rate constants. The mean ratio of the calculated rate constants can be used as a good estimate for the equilibrium constant.

Kinetic models, including MASS models, intrinsically have a layer of uncertainty due to data incompleteness and quality that is estimated. A properly built model minimizes the uncertainty by estimating parameters that have the least effect on network dynamics. Steady-state parametric sensitivity analysis is crucial in determining which equilibrium constants, concentrations, and PERCs can be estimated.

Methods development for dealing with the data incompleteness and quality issue should not be overlooked. New procedures must be developed to account for the intrinsic uncertainty in kinetic parameters.

3.2 Scalability

There are many hurdles for scaling the traditional kinetic approach to larger metabolic pathways. Traditional kinetic models involve analytically deriving rate laws and determining kinetic parameters using in vitro data. This is done case-by-case for each enzyme in a pathway and then the enzymatic rate laws are combined. The process is difficult and time-consuming.

MASS models are data-driven and many of the procedures in building large networks are automated. Genome-scale metabolic reconstructions act as a scaffold for MASS models. The reconstruction process is well-developed and many exist for a variety of prokaryotes and eukaryotes [3, 5, 7, 10]. Metabolomic and fluxomic techniques and data are becoming available for more metabolites and pathways, respectively [2, 24]. The scale-up of MASS kinetic models is more manageable than the traditional approach. However, the increasing size of MASS models lead to a few challenges.

First, central metabolic pathways are well characterized in high-throughput data sets. As MASS models expand to less studied metabolic pathways, data completeness and quality becomes a major issue. Second, computation becomes an issue as the size of the networks increase. Large networks can have ordinary differential equations and matrices that are stiff and ill-conditioned, respectively, thus making large networks harder to dynamically simulate and analyze.

3.3 Condition Specificity

The MASS approach utilizes omics data to hurdle the scalability issues of traditional kinetic modeling. Unlike traditional modeling, kinetic parameters of MASS models are not based on first principles, and thus do not have absolute characteristics. The MASS approach is relative to the steady state under which the omics data is obtained. Potential users of the approach must be weary of this. In addition, there are potential methods to hurdle this problem. Perturbations of the steady state and recalculation of the PERCs can lead to determining kinetic constants that are robust for multiple states and are not condition specific.

4 Red Blood Cell Metabolism

Up to this point in this chapter, we have outlined the reasoning, methodology, and potential challenges of the MASS approach. We now present a full human red blood cell metabolic MASS model in both an unregulated and regulated form adapted from [15].

Table 8.2 General properties and time-scale separation of the unregulated and regulated red cell models

	Red cell model	Regulated red cell model
S dimensions	36×42	92×94
Left null space dimensions	3	10
Right null space dimensions	9	12
G dimensions	42×36	94×92
Time-scale separation	$\sim 20\,\text{min}–5\,\text{h}$	$\sim< \mu s–10\,\text{h}$

4.1 System Definition and Characteristics

The human red blood cell, or erythrocyte, is the most abundant human cell type. It is derived from the hematopoietic stem cell line. Physiologically, erythrocytes take up oxygen from the lungs and deliver it to the tissues. The most abundant protein in erythrocytes is hemoglobin, which is the oxygen carrier molecule. Although the erythrocyte has a crucial physiological role, its metabolism is much simpler to model as compared to other human tissues and cells. Hence, we chose the representation of human red blood cell metabolism as the first whole-cell MASS model. The work accounts for glycolysis, pentose phosphate pathway (PPP), adenosine nucleotide metabolism, the Rapoport-Luebering shunt, and membrane transports and pumps. The size of the stoichiometric, gradient matrices, and null spaces are provided in Table 8.2.

As described earlier in this chapter, the null space determines the steady-state extreme pathways of the system. The pathways represent glycolysis, Rapoport-Luebering shunt, PPP, and the pathways linking nucleotide salvage pathways with PPP. The left null space contains three conserved moieties of the NAD, NADP, and phosphate-containing compounds. This is due to the inability of these molecules to flow in or out of the system. Dynamic analysis was performed to determine the timescale separation in the network, resulting in chemical transformations occurring between 20 min and 5 h.

4.2 Defining Enzyme Modules

The initial red cell model used metabolomic and fluxomic data to estimate PERCs for all the enzymes in the different metabolic pathways. The model ignored regulatory effects. A regulated MASS model was built by integrating five different enzyme modules. The free enzymes and their complexes were added as nodes in the stoichiometric matrix, similar to small metabolites. The mechanism for binding, conversion, and release of substrates to products for all five enzymes was completed similar to (8.13)–(8.19).

The five key regulatory enzymes were hexokinase (HK), phosphofructokinase-1 (PFK), diphosphoglyceromutase (DPGM) and diphosphoglycerol phosphatase (DPGase), glucose-6-phosphate dehydrogenase (G6PDH), and adenosine kinase (AK). HK was modeled using a random-ordered sequential enzyme mechanism accounting for inhibition by glucose-6-phosphate, 2,3-bisphosphoglycerate, and free phosphate. PFK was modeled as a tetramer with the ability to be activated by AMP and inhibited by ATP and magnesium ions. DPGM and DPGase are catalyzed by the same enzyme, utilizing a sequential mechanism with no inhibitors. G6PDH is also represented with a sequential catalytic mechanism. Finally, AK is modeled sequentially with its product AMP as an activator. A schematic for each of the five regulatory enzymes and their stoichiometric integration is shown in Fig. 8.5.

Association and disassociation constants were found in the literature for human erythrocytes. In addition, total concentration for each of the enzymes was known and the flux through the catalytic pathways was assumed to remain the same as the unregulated model. Kinetic rate constants were estimated to be large and used to analytically solve for the enzyme complex concentrations.

4.3 Unregulated vs. Regulated

The regulated and unregulated red blood cell MASS models exhibited starkly different characteristics. First, the regulated model's stoichiometric matrix is much larger (Table 8.2) due to the explicit definition of the enzyme and complex transformations (Fig. 8.5). The left and right null spaces are also affected. There are seven more conserved moieties, corresponding to the enzymes and free magnesium pools. There are three additional pathways, representing the added glycolytic pathways by PFK.

The modal matrix was determined from the Jacobian. Timescale separation for the regulated network is increased ($<\mu$s–10 h). The slow modes represent the metabolite-bound enzyme complexes. Enzyme modules with multiple intermediate steps had the more dominant effect on the slower timescales. It is interesting to note that the time scale separation seen in the regulated model fit closely to the traditional kinetic model for red blood cell metabolism.

The ordinary differential equations for the regulated model were stiff due to the small concentrations of enzyme complexes. Enzyme amounts were normalized and the PERCs were recalculated for proper dynamic simulation. Loads on ATP and NADPH were used to perturb the system for numerical simulation. The regulated model had a more dampened response to pulsed load changes as compared to the unregulated model (Fig. 8.6).

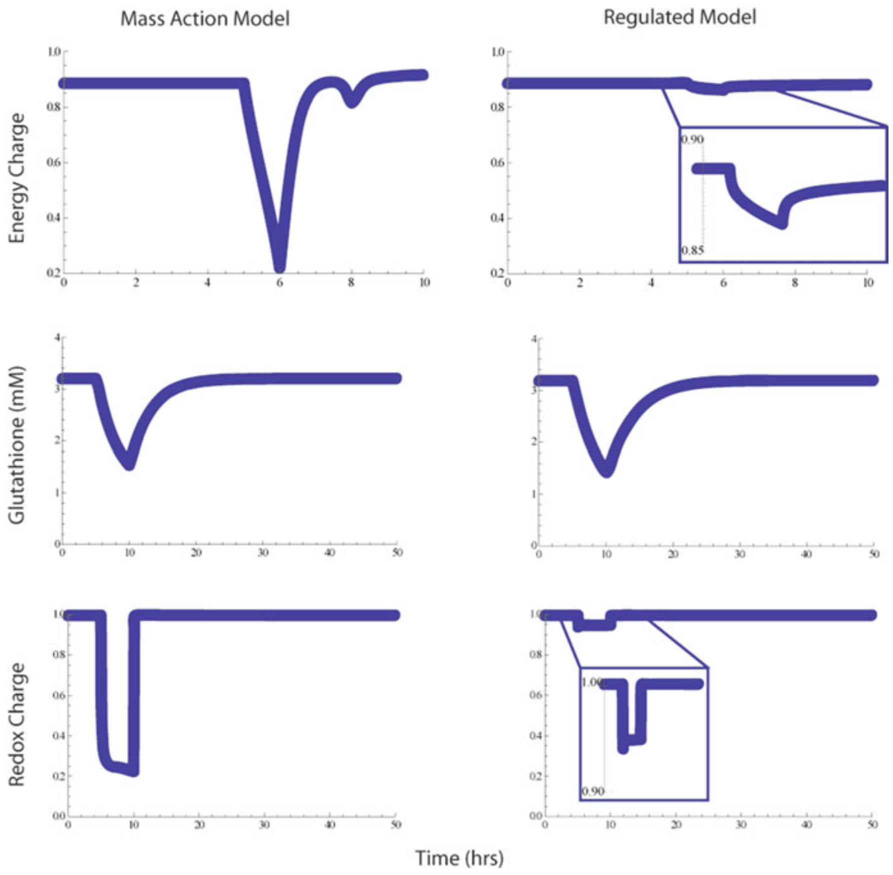


Fig. 8.6 Dynamic responses of the unregulated and regulated red blood cell MASS models. The first row represents a network response to pulsed energy load from t = 5–6 h. The energy charge, $(2^{*}\text{ATP} + \text{ADP})/(2^{*}(\text{ATP} + \text{ADP} + \text{AMP}))$, time profile is shown. The second and third row represent network responses to pulsed redox loads from t = 5–10 h. Redox charge is defined as NADPH/(NADPH + NADP). The regulated network has a dampened response to these perturbations as compared to the unregulated model. Adapted from [15]

5 Conclusions and Future Outlook

Traditional kinetic modeling has been helpful in detailing cellular metabolism. However, traditional methods rely on in vitro enzymatic data and are not scalable. MASS models provide an alternative approach to traditional kinetic modeling efforts of cellular metabolism. There has been advancement of holistically characterizing living systems with high-throughput data sets. This new field, called omics, provides the necessary data to parameterize MASS models.

In this chapter, we provided an outline on how to use a metabolic network, top-down measurements, and equilibrium constants to build dynamic mass balances for kinetic metabolic networks on a large scale. A process was elucidated for explicitly characterizing enzymes and allosteric regulatory effects within the MASS framework. Analytical methods were introduced to reduce the dimensionality of the network, determine the time-scale separation of the biochemical transformations, and simulate the dynamics of the metabolic system dealing with perturbations. Although the workflow for building and analyzing MASS models has been outlined, some challenges still remain. Kinetic models, including MASS, cannot escape the data incompleteness and quality issue. New methods need to be developed to deal with uncertainty in kinetic parameters.

Full red blood cell metabolism can be modeled in the MASS framework. Enzyme modules are defined and integrated with the metabolite stoichiometric matrix to allow for explicit definition of allosteric regulation. The regulated model exhibited time-scale separation reminiscent of traditional kinetic models. Building the full red blood cell MASS model with regulation taught us a few things. First, the MASS approach is scalable and more complex cellular models are possible. Second, explicit definition of allosteric enzyme regulation in a modular format is feasible and provides a better representation of dynamic metabolic properties. Third, new methods are required to determine uncertainties in the network.

We see the MASS approach and its accompanying methods to follow a similar path as COBRA based models and methods. MASS models are data-driven and can be expanded to genome-scale. There are two key areas that need to be developed side-by-side for the success of expanding MASS models.

First, more complex cells need to be modeled, including explicitly defining and adding more regulatory enzymes to the network to account for regulation. Phospho-proteomic data is now becoming available and allows accounting for the phosphorylated states of enzymes. With emerging reconstructions of transcription regulatory networks [11], transcription of enzymes can be explicitly modeled using ordinary differential equations.

Second, building more complex networks will define many of the practical issues of the MASS approach. We foresee parallel sampling of kinetic parameters and unknown concentrations to be of great interest to MASS models. Other areas that require attention include: estimating unknown equilibrium constants, determining meaningful steady-state pathways from the null space, and dealing with the computation of ill-conditioned matrices and stiff ordinary differential equations.

Metabolism plays an important role in a number of major human diseases including cancer. Kinase deficiency has been shown to play a major role in oncogenesis [22]. Constraint-based models have been helpful in understanding some biochemical mechanisms of pathogens [13, 20]. However, human metabolism has an inherent dynamic behavior that cannot be accounted for in constraint-based models. Enzyme states affected by allosteric ligands, phosphorylation, and transcription are also ignored in constraint-based and traditional kinetic models. The MASS approach is a scalable method to model cellular metabolism while accounting for dynamic network properties and enzymes.

References

1. Becker SA, Feist AM, Mo ML, Hannum G, Palsson BO, Herrgard MJ (2007) Quantitative prediction of cellular metabolism with constraint-based models: The COBRA Toolbox. Nat Protocols 2: 727–738.
2. Bennett BD, Yuan J, Kimball EH, Rabinowitz JD (2008) Absolute quantitation of intracellular metabolite concentrations by an isotope ratio-based approach. Nature protocols 3: 1299–1311.
3. Duarte NC, Becker SA, Jamshidi N, Thiele I, Mo ML, Vo TD, Srivas R, Palsson BO (2007) Global reconstruction of the human metabolic network based on genomic and bibliomic data. Proceedings of the National Academy of Sciences of the United States of America 104: 1777–1782.
4. Edwards JS, Palsson BO (1999) Systems properties of the Haemophilus influenzae Rd metabolic genotype. Journal of Biological Chemistry 274: 17410–17416.
5. Famili I, Forster J, Nielsen J, Palsson BO (2003) Saccharomyces cerevisiae phenotypes can be predicted by using constraint-based analysis of a genome-scale re-constructed metabolic network. Proceedings of the National Academy of Sciences of the United States of America 100: 13134–13139.
6. Famili I, Mahadevan R, Palsson BO (2005) k-Cone Analysis: Determining All Candidate Values for Kinetic Parameters on a Network Scale. Biophysical journal 88: 1616–1625.
7. Feist AM, Henry CS, Reed JL, Krummenacker M, Joyce AR, Karp PD, Broadbelt LJ, Hatzimanikatis V, Palsson BØ (2007) A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. Molecular systems biology 3: 121.
8. Feist AM, Palsson BO (2008) The growing scope of applications of genome-scale metabolic reconstructions using Escherichia coli. Nat Biotech 26: 659–667.
9. Feist AM, Zielinski DC, Orth JD, Schellenberger J, Herrgard MJ, Palsson BO (2010) Model-driven evaluation of the production potential for growth-coupled products of Escherichia coli. Metabolic engineering 12:3 173–186.
10. Forster J, Famili I, Fu PC, Palsson BO, Nielsen J (2003) Genome-Scale Reconstruction of the Saccharomyces cerevisiae Metabolic Network. Genome Research 13: 244–253.
11. Gianchandani EP, Joyce AR, Palsson BO, Papin JA (2009) Functional states of the genome-scale Escherichia coli transcriptional regulatory system. PLoS compu-tational biology 5: e1000403.
12. Irani MH, Maitra PK (1977) Properties of Escherichia coli mutants deficient in enzymes of glycolysis. Journal of bacteriology 132: 398–410.
13. Jamshidi N, Palsson BO (2007) Investigating the metabolic capabilities of Mycobacterium tuberculosis H37Rv using the in silico strain iNJ661 and proposing al-ternative drug targets. BMC systems biology 1: 26.
14. Jamshidi N, Palsson BO (2008) Top-down analysis of temporal hierarchy in biochemical reaction networks. PLoS computational biology 4: e1000177.
15. Jamshidi N, Palsson BO (2010) Mass Action Stoichiometric Simulation Models: Incorporating Kinetics and Regulation into Stoichiometric Models. Biophysical journal 98: 175–185.
16. Lueck JD, Fromm HJ (1974) Kinetics, mechanism, and regulation of rat skeletal muscle hexokinase. The Journal of biological chemistry 249: 1341–1347.
17. Maitra PK, Lobo Z (1971) A kinetic study of glycolytic enzyme synthesis in yeast. The Journal of biological chemistry 246: 475–488.
18. Oberhardt MA, Palsson BO, Papin JA (2009) Applications of genome-scale metabolic reconstructions. Molecular systems biology 5: 320.
19. Palsson BO (2006) Systems biology: properties of reconstructed networks. Cambridge University Press, New York.
20. Raghunathan A, Reed J, Shin S, Palsson B, Daefler S (2009) Constraint-based analysis of metabolic capacity of Salmonella typhimurium during host-pathogen interaction. BMC systems biology 3: 38.

21. Thiele I, Palsson BO (2010) A protocol for generating a high-quality genome-scale metabolic reconstruction. Nature protocols 5: 93–121.
22. Vander Heiden MG, Cantley LC, Thompson CB (2009) Understanding the War-burg effect: the metabolic requirements of cell proliferation. Science, New York, NY 324: 1029–1033.
23. Westerhoff HV, Palsson BO (2004) The evolution of molecular biology into systems biology. Nature biotechnology 22: 1249–1252.
24. Yuan J, Bennett BD, Rabinowitz JD (2008) Kinetic flux profiling for quantitation of cellular metabolic fluxes. Nature protocols 3: 1328–1340.

Index

A

Additive coordinate dysregulation, transcriptomics, 182–183
Adjacency matrix, 17, 99, 163, 164
Alignment graph, 32, 101–103, 110, 117
Alternative splicing, 2, 189
Annotation. *See* Functional annotation, in gene networks
Annotation clustering, 57
Arabidopsis Information Resource (TAIR), 122
Ascertainment bias, 4
Automorphism, 21, 23

B

Bayes' rule, 174
Betweenness distribution, 78–79
Between-pathway model, 180
Biclustering algorithm, 149–150
BioCyc database, 121
Biological General Repository for Interaction Datasets (BioGRID), 120
Biological network alignment
  applications
    conserved functional modules, 123–124
    disease discovery, 126–128
    evolutionary events, 124–126
    projecting functional annotations, 122–123
  databases, 119–122
  definitions and notations
    alignment scoring function, 101
    graphs, 99
    metabolic networks, 100
    node scoring function, 101
    pairwise problem, 100–101
    synthetic lethality, 100
  global alignment
    Graemlin family, 111–113
    integer quadratic programming formulations, 113–116
    IsoRank family, 110–112
    Markov random field, 110
  vs. graph kernels, 130–131
  vs. graph matching, 129–130
  issues of, 98, 99
  local alignment
    Blast family, 102–104
    de novo, 109
    Graemlin, 106–108
    information theoretic network alignment, 108–109
    MAWISH, 104–106
    network queries, 109
  multiple alignment, complexity and scalability, 116–117
  and network analyses problems, 129
  subgraph isomorphism problem, 98
  validation methods
    consistency and coverage, 119
    functional coherence, 118, 119
    ontologies, 117
    sensitivity and specificity, 119
    similarity score, 118
Biological networks, pattern mining
  classification, 138
  coexpression network, 138
  coherent dense subgraphs, recurrent patterns
    aggregated graph, 139
    CODENSE algorithm, 141–143
    experimental study, 143–144
    problem formulation, 139–142

M. Koyutürk et al. (eds.), *Functional Coherence of Molecular Networks in Bioinformatics*, DOI 10.1007/978-1-4614-0320-3,
© Springer Science+Business Media, LLC 2012

Biological networks, pattern mining (*cont.*)
- summary graph, edge occurrence profiles, 144
- differential network patterns
  - annealing schedule, 157
  - cluster homogeneity, 158, 159
  - coexpression graphs, 155–156
  - conceptual homogeneity, 159
  - density differential function, 154
  - functional homogeneity, 158
  - GeneRIF, 160
  - GO functions, 159, 160
  - individual objective functions, 154
  - initial states, 156
  - neighboring states, selection of, 156–157
  - phenotype, 153, 158, 160, 162
  - postfiltering, 157
  - problem formulation, 153–155
  - search space, 155
  - statistical significance, 162
  - UMLS concepts, 158, 160, 161
- frequent dense vertexset (FDVSs), recurrent patterns
  - algorithm, 146–147
  - ChIP-chip and GO data, 148
  - experimental study, 148–149
  - neighbor association summary graph, 147
  - problem formulation, 145–146
  - summary graph approach, 145, 146
- general recurrent network patterns
  - biclustering algorithm, 149–150
  - experimental study, 151–152
  - gene functions prediction, 150–151
- multiple large networks, 162
- tensor, computational model for
  - continuous optimization techniques, 163
  - experimental study, 166–167
  - heavy subgraph (HS), 163
  - problem formulation and optimization algorithm, 164–166
  - recurrent heavy subgraph (RHS), 163–167
  - third-order tensor, coexpression networks, 164

Biological process, 50, 53, 55, 143, 160

Biomolecular Interaction Network Database (BIND), 119–120

Bipartite graph, 113, 114

Blast family
- logistic regression model, 103
- log-likelihood ratio of, 104
- match, gap/mismatch edge, 102
- NetworkBlast and NetworkBlast-M, 103
- PathBlast, 102, 103
- probability of true interaction, 103
- scoring function, 102

C

Caenorhabditis elegans, 32, 64, 103, 123, 124, 127

Candidate gene, 174, 175, 190

Catalytic pathways, 210, 216

Cellular component, 2, 53, 166

Clique, 81, 85, 139, 144, 163

Closeness distribution, 79

Clustering coefficient, 20

Clustering spectrum, 20, 21, 23

Clusters of Orthologous Groups (COG), 107, 123

Coherent dense subgraphs, biological network
- aggregated graph, 139
- CODENSE algorithm, 141–143
- experimental study, 143–144
- problem formulation, 139–142
  - edge support vector, 140
  - second-order graph, 140
- summary graph, edge occurrence profiles, 144

Colorectal cancer, TCP1 subnetwork, 194–195

Combinatorial coordinate dysregulation, transcriptomics, 185–187

Comparative network analysis, 8, 117, 119, 123–125, 127

Complementarity, of genes, 185

Computational annotation
- direct approaches, 54
- interaction networks, 54
- module-assisted approaches, 55, 56
- pattern-based approaches
  - Indirect Neighbor, 56
  - LaMoFinder, 56
  - Similar Neighborhood and Annotation Clustering, 57
- probabilistic approaches, 55, 56

Computational function prediction and gene networks, 50–52

Conceptual network, 138

Connectivity, 37, 78, 82, 125

Constraint-based reconstruction and analysis (COBRA), 202

Coordinate dysregulation
- additive, 182–183
- combinatorial, 185–187

D
DADA, 179
Database of Interacting Proteins (DIP), 120
Databases, 119–122
Degree distribution, 20, 78
Dense subgraph. *See* Coherent dense subgraphs, biological network
Diameter, 20, 25–27
Divergence, 74, 105
DNA Data Bank of Japan (DDBJ), 121
Drosophila melanogaster, 32, 103, 115, 122, 123, 127
Dysregulated pathway, 180
Dysregulated subnetworks, proteomics. *See* Proteomics, dysregulated subnetworks

E
Edge support vector, 140
Eigenvalue, 73, 112, 209
Eigenvector, 32, 73
Eigenvector centrality, 24
Emulating models. *See* Proteome network emulating models
Entropy, 119, 182
Enzyme Commission (E.C.), 117
Enzyme modules, 215–216
Enzymes, 210, 211
Equilibrium constant, 204, 206, 213
Erdös–Rényi random graph model, 78, 83
Escherichia coli, 83, 90, 91, 103
European Bioinformatics Institute (EBI), 121, 152
Evolutionary distance, 104, 106

F
Feature vector, 111, 126
Filtered yeast interactome (FYI), 61, 62
Flow cytometry, 190
Flybase database, 122
Frequent dense vertexset (FDVSs), biological network
  algorithm, 146–147
  ChIP-chip and GO data, 148
  experimental study, 148–149
  neighbor association summary graph, 147
  problem formulation, 145–146
  summary graph approach, 145, 146
Functional annotation, in gene networks
  accuracy comparison, of methods
    filtered yeast interactome (FYI) and BioGRID, 61, 62
    K-Nearest-Neighbors (KNN), 61–63
    true positive (TP) *vs.* false positive (FP), 61, 62
  computational function prediction
    Gene Expression Omnibus (GEO), 52
    Gene Ontology (GO), 50–51
  multiple genome networks, hypothesis in, 63–65
  network annotation prediction, as framework
    advantages, 58
    classification, 60–61
    feature extraction, 59–60
    function's score, 60
    interaction network synthesis, 58
    neighborhood patterns, 58
    steps, 57
  network synthesis
    annotation domains, GO, 53
    functional interaction, 52–53
    genetic interactions, 52
    phylogenetic profiling, 53
    protein–protein physical interactions, 52
  techniques, for computational annotation
    direct approaches, 54
    interaction networks, 54
    module-assisted approaches, 55, 56
    pattern-based approaches, 56–57
    probabilistic approaches, 55, 56
Functional flow, 36
Functional modules, 123–124
Functional profile, 59, 60, 64–65
Futile cycle, 213

G
GenBank, 121
Gene duplication-based models, 73–77
Gene Expression Omnibus (GEO), 52
Gene Ontology (GO), 50–51, 53, 159, 160
  hierarchy, 118, 144
Gene Reference Into Function (GeneRIF), 160
Gene regulatory networks, 6–7
Genome-scale kinetic models. *See* Mass action stoichiometric simulation (MASS)
Genomics
  disease genes, interactions, 173–175
  information-flow-based methods
    missing interactions, 174
    multiple paths, 175, 176
    network propagation, 177
    random walk with restarts (RWR), 175–177

Genomics (*cont.*)
mechanistic bases, of genetic effects, 180
network centrality, role of, 178–179
statistical correction, 178, 179
Geometric random graph model, proteome network
embedding of, 72, 73
gene duplication-based models
Ohno's theory, of genome evolution, 74
parameter selection, in generalized model, 76–77
Pastor–Satorras model, 74, 75
pure duplication model, 75
random edge addition, 74–75
limitations of, 72
symmetric and positive semi-definite matrix, 73
two dimensional, 72
Gibbs sampling, 110
Global network alignment, 32–34
Graemlin family, 111–113
integer quadratic programming formulations, 113–116
IsoRank family, 110–112
Markov random field, 110
Gradient descent, 116, 166
Gradient matrix, 209
Graemlin
equivalence classes
edge scoring matrix (ESM), 107
progressive alignment, 108
scoring function, 107
seed extension, *d*-clusters, 107, 108
hill-climbing method, 113
loss function, 112, 113
weight function, 112
Graph density, 140
Graph isomorphism, 77
Graph kernels, 130–131
Graphlet degree distribution, 23
Graphlet degree vector (GDV), 21, 24
Graph matching, 129–130

H
Haemophilus influenza, 202
Helicobacter pylori, 83, 90, 91, 124
Homo sapiens, 103, 123, 127
Hungarian algorithm-based GRAAL (H-GRAAL), 33

I
Immunoblot, 194
Information theoretic network alignment
conserved regions, 109
neighborhood conservation, 108
relative description length (RDL), 108, 109
shortest path conservation, 108
Information theory, 108
In-paralog, 123, 125
In silico computational predictions, of interactions, 61
IntAct database, 120
Integer quadratic programming (IQP) formulations
bipartite graph, 113, 114
definition, 114–116
maximum weight matching, 114
Integrin, 188, 189
International Molecular Exchange Consortium (IMEx), 120
International Nucleotide Sequence Database (INSD), 121
In vitro enzyme data, 202, 217
In vivo data, 202
IsoRank family, 110–112

J
Jacobian matrix (J), 209

K
K-Nearest-Neighbors (KNN), 61–63
Kyoto Encyclopedia of Genes and Genomes (KEGG), 120

L
LaMoFinder, 56
Local network alignment, 31–32
Blast family, 102–104
gaps, 31
Graemlin, 106–108
information theoretic network alignment, 108–109
MAWISH, 104–106
network queries, 109

M
Markov random field (MRF), 110
Mass action stoichiometric simulation (MASS)
challenges in
condition specificity, 214
data completeness and quality, 212–213
scalability, 214

steady-state flux state, 213
development of, 218
metabolism, 201, 218
metabolites, 204, 205, 210
model construction and analysis
aggregation of concentrations, into pools, 206–208
data sources and requirements, 203–205
dynamic analysis, 208–210
dynamic network properties, analytical methods for, 207
dynamic simulation, 208
enzymes, 210, 211
Jacobian matrix (J), 209
null spaces and steady state, 205–206
omics data types, 205
overview of, 204
pseudo-elementary rate constants, 206
QC/QA procedures, 211–212
regulated red blood cell model, topological overview of, 211
workflow of, 205
omics, 202–203
red blood cell metabolism
enzyme modules, 215–216
models, properties and time-scale separation of, 215
system definition and characteristics, 215
unregulated *vs.* regulated, 216, 217
traditional kinetic modeling, 202, 203
Mass spectrometry (MS), 3, 16, 61, 190
Matrix model, 17, 18
Maximum common subgraph, 130
Maximum weight bipartite matching, 114
Maximum weight induced subgraph (MaWISh), 32, 104–106
Metabolic networks, 5–6
Metabolic pathway, 4, 5, 214
Metabolomics, 205
MetaCyc, 5, 121
Microphthalmia, 179
MIPS, 61, 118
Molecular function, 50, 51, 53, 55
Molecular INTeraction database (MINT), 120
Molecular interaction networks
gene regulatory networks, 6–7
metabolic networks, 5–6
protein–protein, 3–5
signal transduction pathways, 7
Molecular networks and complex diseases, 172
disease gene prioritization, 174–176, 179, 191
genomics
disease genes, interactions, 173–175
information-flow-based methods, 174–177
linkage interval, 174, 175
mechanistic bases, of genetic effects, 180
network centrality, role of, 178–179
omic datasets, 195
proteomics
dysregulated subnetworks, 190–194
TCP1 subnetwork, post-transcriptional dysregulation of, 194–195
vs. transcriptomic, 189–190
transcriptomics, dysregulated subnetworks
additive coordinate dysregulation, 182–183
combinatorial coordinate dysregulation, 185–187
coordinate dysregulation and cover, 183, 184
differential expression, 182
dysregulated genes to, 181–182
gene expression and molecular network data, integration of, 181
gene expression, differential analysis of, 181
positive and negative cover set, 183
precision and recall, 187
subnetwork markers, biological insights, 187–189
synergistic dysregulation, 184–185
MOPHY, 126
Mouse Genome Database (MGD), 122
MPact database, 120
mRNA degradation, 189
mRNA expression, 189, 190
Multiple genome networks, 63–65
Multiple network alignment, 116–117
Multi-stage convex relaxation (MSCR) optimization, tensor, 166
Mus musculus, 103, 123
Mutual information phenotype, 184, 185

N

NCI-Nature Pathway Interaction Database (PID), 121
Netpath database, 121
Network alignment, 30
global
GRAAL, 33
H-GRAAL, 33
IsoRank, 32
local

Network alignment (*cont.*)
    MaWISh, 32
    PathBLAST, 31
Network analysis
  global properties
    clustering coefficient, 20
    defined, 19, 20
    degree distribution, 20
    graphlets, 21
    shortest path, 20
    small-world property, 20
  isomorphism, 19
  local properties
    graphlets, 23
    network fingerprint, 24
    network motifs, defined, 22
  node centrality, 24
Network annotation prediction, as framework
  advantages, 58
  classification, 60–61
  feature extraction
    neighborhood profile, transformation of, 59, 60
    power iteration approach, stationary distribution, 59
  interaction network synthesis, 58
  steps, 57
Network biology
  molecular interaction networks
    gene regulatory networks, 6–7
    metabolic networks, 5–6
    protein–protein, 3–5
    signal transduction pathways, 7
  molecular networks
    and biological function, 7–9
    functional coherence, 9
  systems biology, 1–2
NetworkBlast, 103
Network centrality, 178–179
Network degree, 178, 179
Network feature extraction, 59–60
Network modeling
  biological applications, 29–30
  Erdös–Rényi random graph, 25
  geometric random graph, 26
  scale-free network, 26, 27
  small-world network, 26, 27
  stickiness network model, 28
Network motifs, defined, 22
Network propagation, 177
Network scheme, 109
Network topology, 9, 19, 33–38
Neural networks (NNs), 187
Noise, 18, 145
NP-complete, 19, 77, 80, 101
NP-hard, 98, 104, 138

O
Omics, 202–203
Online Mendelian Inheritance in Man (OMIM), 118, 122, 173, 177–179
Orbits, 21, 23, 24
Orthologs, 64
Out-paralogs, 110, 123

P
2D-PAGE, 189
PageRank, 32, 111
Paralogs, 107, 123
Partial subgraph, 22
Pastor–Satorras model, proteome network, 74, 75
PathBlast, 31, 102, 103
Phospho-proteomics, 218
Phylogenetic tree, 34, 117, 127
Physical network, 138
Post-translational abundance, 190
Post-translational modifications, 2, 190
Power law, 20, 27, 70, 71, 74, 78
Preferential attachment model, proteome network, 70, 83, 87
Protein expression, 189–195. *See also* Proteomics
Protein Information Resource (PIR), 121
Protein–protein interaction (PPI) network. *See also* Proteome network emulating models
  biological function
    clusters, 36
    hubs, 35
    protein function prediction, 36
  disease and pharmacology, 37–38
  molecular interaction networks, 3–5
Protein Sequence Database (PSD), 121, 122
Proteome network emulating models, 93
  geometric random graph model
    embedding of, 72, 73
    gene duplication-based models, 73–77
    limitations of, 72
    symmetric and positive semi-definite matrix, 73
    two dimensional, 72
  network evolution models, assessment of
    betweenness distribution, 78–79
    challenges, 90–92
    closeness distribution, 79

degree distribution, 78
dense graphlets, 92
Erdös–Rényi model, topological properties of, 83, 86
features, robustness of, 81–84
generalized duplication model, seed network selection, 85, 89, 90
graph isomorphism, 77
graphlet, 79–81
k-hop reachability, 78
perturbation, 82, 83
sizes, comparison of, 90, 91
subgraph frequency, 79–81
treelets, 80, 82, 91
vertices, edges, and average degree, number of, 85
yeast network, preferential attachment model, 83, 87
yeast network, random geometric model, 85, 88
preferential attachment models, 71
random graph models, 70
yeast, 2-hybrid assays, 70
Proteomics
dysregulated subnetworks
candidate subnetworks in, 193
colorectal cancer (CRC), 190
complex phenotypes, framework for, 191, 192
crosstalk, 191
crosstalker subnetworks, 191
interactor subnetworks, 191
proteomic seeds, 191, 193
proteomic targets and mRNA-level dysregulation, 193
TCP1 subnetwork, post-transcriptional dysregulation of, 194–195
vs. transcriptomic, 189–190
Pseudo-elementary rate constants (PERCs), 206

Q
Query subnetwork, 109

R
Random graph, 25, 26
Random graph model. *See* Geometric random graph model, proteome network
Random walk, 131, 151
Random walk with restarts (RWR), 175–177
Rat Genome Database (RGD), 122
Reactome database, 121
Red blood cell metabolism, MASS approach
enzyme modules, 215–216
models, properties and time-scale separation of, 215
system definition and characteristics, 215
unregulated *vs.* regulated, 216
dynamic responses of, 217
regulated model, topological overview of, 211
Redundancy, of genes, 185
RNA interference, 189

S
Saccharomyces cerevisiae, 16, 115, 120, 124, 127
Saccharomyces Genome Database (SGD), 122
Seed graph, 70, 85
Signaling pathway, database, 121
Small world effect, 70
Spoke model, 18, 35
Stochastic matrix, 112, 151, 176
Stoichiometric matrix, 204–206, 211, 212
Subgraph centrality, 24
Support vector machines (SVMs), 187
Swiss-Prot, 121, 122
Synergistic dysregulation, transcriptomics, 184–185
Synergy, 184, 185
Synergy network, 185
Synthetic lethal interaction, 180
Systems biology, defined, 1–2
Systems pharmacology, 38

T
Tandem affinity purification (TAP), 3, 4
Tensor, multiple weighted networks
experimental study, 166–167
problem formulation and optimization algorithm
multi-stage convex relaxation (MSCR) optimization, 166
recurrent heavy subgraph (RHS), 164, 165
vector norm constraints, 165
Topological characteristics
biological networks
datasets, 16
modeling, 18
nodes, 15–16
PPI, 16–17
network alignment
global, 32–34

Topological characteristics (*cont.*)
        local, 31–32
    network analysis
        global properties, 19–22
        local properties, 22–24
        node centrality, 24
    network modeling
        biological applications, 29–30
        models, network, 25–29
    PPI networks, 35–36
        biological function, 35–36
        disease and pharmacology, 37–38
TOpology-free netwoRk QUErying (Torque), 109
Transcriptional regulation, 6
Transcriptomics, dysregulated subnetworks
    additive coordinate dysregulation, 182–183
    combinatorial coordinate dysregulation
        vs. additive, 185, 186
        CRANE, colon cancer metastasis, 187
        individual subnetwork state functions, 186
    coordinate dysregulation and cover, 183, 184
    differential expression, 182
    dysregulated genes to, 181–182
    gene expression
        differential analysis of, 181
        and molecular network data, integration of, 181
    subnetwork activity, 182
    subnetwork markers, biological insights
        hypothesis-driver subnetwork, interaction diagram, 187, 188
        MMP proteins, 188
    synergistic dysregulation, 184–185
Transcriptomic *vs.* proteomic data, 189–190

U
UMLS, 158, 160, 161

V
Vertexlet, 147
Vitronectin, 188

W
Within-pathway models, 180

Y
Yeast proteome network
    2-hybrid assays, 70
    preferential attachment model, 83, 87
    random geometric model, 85, 88