COMPARATIVE ANALYSIS OF BIOLOGICAL NETWORKS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Mehmet Koyutürk

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2006

Purdue University

West Lafayette, Indiana

"*bilmezler nasıl aradık birbirimizi,*
bilmezler nasıl sevdik,
iki yitik hasret,
iki parça can..."

to Günnur...

ACKNOWLEDGMENTS

I am grateful to Ananth Grama, who contributed a great deal in my academic and professional development, by sharing his broad knowledge, deep insight, and encouraging optimism. He was more than supportive and understanding in every aspect of research and life. This dissertation represents only a small part of what I gained from working with him throughout all these years.

It was great joy and an invaluable learning experience to work with Wojciech Szpankowski, who continuously supported and encouraged me with his never-ending enthusiasm and energy. He taught me how to "nail the beast down!" by patiently struggling against problems side by side with me. He will always be an inspirational role model with his enlightening wisdom and "curious mind".

Our fruitful collaboration with Shankar Subramaniam constructed the backbone of this study. This thesis would never come into existence without his inquiring mind, which provided us with many research directions, and his biological wisdom, which gave strength to our algorithms.

I would like to thank the members of my thesis committee, Alberto Apostolico, Daisuke Kihara, and Robert Skeel, for reading this dissertation, being always available, and providing valuable feedback. William Gorman was very helpful in putting this dissertation into a good shape.

Most of the work presented here is the result of collaboration with Yohan Kim. He contributed in various ways, including intense exchange of ideas, evaluation and discussion of results, and critical reading, strengthening this work significantly.

I appreciate the friendship and contribution of my colleagues Ioannis Ioannidis, Bogdan Cárbunar, Ronaldo Ferreira, Jie Chi, Wenhui Ren, Muralikrishna Ramanathan, Deepak Bobbarjung, Asad Awan, Umut Topkara, Metin Aktulga, Jayesh Pandey, and Sagar Pandit. They were always available for rewarding discussions and

generous in sharing ideas. I would also like to thank Mercan Topkara and Murat Kantarcıoğlu, who shared the load of being a graduate student with me.

Life during grad school was delightful and unforgettable, thanks to Grace-Brad Sheese and Pınar-Hakkı Gürkaş. They turned *gurbet*[1] into *sıla*[2] for us.

Our families were always supportive, understanding, and available. There is no way I can describe my mother's contribution in words. She and my father put a lot of effort and did not hesitate to sacrifice anything to provide us a good education and future. I have always been lucky to have two great sisters, Dilek and Belgin, who contributed a lot in my personal and intellectual development. My brothers, sisters, and parents in law always shared their love and support; many thanks to Alp for showing great interest in my work and listening to me. My mother in law has been inspring with her strength and wisdom. Our nieces and nephews, Berk, Gün, Onat, İrem, Beliz, and Tan brought joy and energy to our life.

Finally, I am grateful to my better half, Günnur, for always being with me, listening, understanding, and encouraging me openheartedly, and being always there to help.

[1] *Tr.* A location away from home. Where one is a stranger.

[2] *Tr.* Home. Reunion. Where one belongs to.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Koyutürk, Mehmet Ph.D., Purdue University, December, 2006. Comparative Analysis of Biological Networks. Major Professors: Ananth Grama and Wojciech Szpankowski.

Recent developments in molecular biology have resulted in experimental data that entails the relationships and interactions between biomolecules. Biomolecular interaction data, generally referred to as biological or cellular networks, are frequently abstracted using graph models. In systems biology, comparative analysis of these networks provides understanding of functional modularity in the cell by integrating cellular organization, functional hierarchy, and evolutionary conservation. In this dissertation, we address a number of algorithmic issues associated with comparative analysis of molecular interaction networks.

We first discuss the problem of identifying common sub-networks in a collection of molecular interaction networks belonging to diverse species. The main algorithmic challenges here stem from the exponential worst-case complexity of the underlying mining problem involving large patterns, as well as the NP-hardness of the subgraph isomorphism problem. Three decades of research into theoretical aspects of this problem has highlighted the futility of syntactic approaches to this problem, thus motivating use of semantic information. Using a biologically motivated ortholog-contraction technique for relating proteins across species, we render this problem tractable. We experimentally show that the proposed method can be used as a pruning heuristic that accelerates existing techniques significantly, as well as a stand-alone tool that conveys significant biological insights at near-interactive rates.

With a view to understanding the conservation and divergence of functional modules, we also develop network alignment techniques, grounded in theoretical models

of network evolution. Through graph-theoretic modeling of evolutionary events in terms of matches, mismatches, and duplications, we reduce the alignment problem to a graph optimization problem and develop effective heuristics to solve this problem efficiently.

We probabilistically analyze the existence of highly connected and conserved subgraphs in random graphs, in order to assess the statistical significance of the patterns identified by our algorithms. Our methods and algorithms are implemented on various platforms and tested extensively on a comprehensive collection of molecular interaction data, illustrating the effectiveness of the algorithms in terms of providing novel biological insights as well as computational efficiency. The source code of the software described in this dissertation is available in the public domain and has been downloaded and effectively used by several researchers.

1. INTRODUCTION

Increasing availability of experimental data relating to biological sequences coupled with efficient tools such as BLAST and CLUSTAL have contributed to fundamental understanding of a variety of biological processes [1, 2]. These tools help in understanding relationships as well as differences between sequences and associated organisms. Common subsequences and motifs discovered by such tools are used to derive functional, structural, and evolutionary information.

Recent developments in molecular biology have resulted in a new generation of experimental and computational data that entails the relationships and interactions between biomolecules [3]. With the availability of high-throughput screening methods [4–7] and computational prediction techniques [8, 9], interaction data for various organisms is available in the form of several abstractions. These abstractions include metabolic pathways, protein-protein interaction (PPI) networks, and gene regulatory networks. These abstractions and associated data facilitate understanding of cellular organization in a systems framework. These networks are organized in public databases, which provide simple search queries as well as bulk data downloads making molecular interaction data available and accessible to a broad class of researchers [10–12]. Although vast amounts of data is becoming increasingly available, efficient analysis counterparts to BLAST and CLUSTAL are not readily available for such abstractions.

Biomolecular interaction data, generally referred to as biological or cellular networks, are frequently abstracted using graph models [13, 14]. As is the case with sequences, two key problems on graphs are: aligning multiple graphs, and finding frequently occurring subgraphs in a collection of graphs. Analysis of biological networks in terms of these problems provides understanding of several biologically interesting concepts such as common motifs of cellular interactions, evolutionary relationships

and differences among cellular network structures of different organisms, organization of functional modules, relations and interactions between sequences, and patterns of gene regulation. In this study, we develop algorithms for discovering conserved substructures in a collection of networks and alignment of interaction networks, and provide statistical analyses for assessing the significance of conserved and densely connected subgraphs in biological networks.

Preliminary studies on molecular interaction data show that functional conservation is likely to manifest itself in terms of conservation of interactions [15]. Such observations, coupled with the availability of interaction data for tens of species, motivate comprehensive investigation of conserved substructures in interaction networks belonging to a diverse range of species. In terms of the graph-theoretical abstraction of biological networks, the corresponding computational problem can be described as one of identifying common (frequent) subgraphs in a collection of graphs.

The main algorithmic challenges in identification of conserved substructures stem from the exponential worst-case complexity of the underlying mining problem involving large patterns, as well as the NP-hardness of the subgraph isomorphism problem. Three decades of research into theoretical aspects of this problem has highlighted the futility of syntactic approaches to this problem – thus motivating use of semantic information. Using an innovative graph simplification technique based on ortholog contraction, which is ideally suited to biological networks, we develop an algorithm that renders these problems computationally tractable and scalable to large numbers of networks. We show, experimentally, that the resulting software, MULE[1], can extract frequently occurring patterns in metabolic pathways and PPI networks collected from several databases within seconds [16]. When compared to existing approaches, our graph simplification technique can be viewed either as a pruning heuristic, or a closely related, but computationally simpler task [17]. When used as a pruning heuristic, our technique reduces effective graph sizes significantly, accelerating existing techniques

[1]The source code of MULE is publicly available at `http://www.cs.purdue.edu/homes/koyuturk/pathway` and is already downloaded by more than a hundred researchers.

by several orders of magnitude! Indeed, for most of the available networks, existing techniques can not even be applied without our pruning step. When used as a stand-alone analysis technique, MULE is shown to convey significant biological insights at near-interactive rates.

Detection of conserved interaction patterns on a collection of biological networks may be thought of as the counterpart of multiple sequence alignment in the network domain. Another important and useful class of tools in comparative genomics is pairwise sequence alignment, which makes it possible to search for sequences similar to a query sequence in a database of protein or DNA sequences. Similarly, pairwise network alignment is promising in terms of understanding the conservation and divergence of interactions and modular tasks in two different species, as well as searching for "orthologs" of a given functional module or protein complex in a large network database.

The main challenge in PPI network alignment is to define a graph theoretical measure of similarity between graph structures that captures underlying biological phenomena accurately. In this respect, modeling of conservation and divergence of interactions, as well as the interpretation of resulting alignments are important design parameters. We develop a framework for comprehensive alignment of PPI networks, which is inspired by duplication/divergence models that focus on understanding the evolution of protein interactions. We propose a mathematical model that extends the concepts of match, mismatch, and gap in sequence alignment to that of match, mismatch, and duplication in network alignment, and evaluates similarity between graph structures through a scoring function that accounts for evolutionary events [18].

By relying on evolutionary models, the proposed framework facilitates interpretation of resulting alignments in terms of not only conservation but also divergence of modularity in PPI networks. Furthermore, as in the case of sequence alignment, our model allows flexibility in adjusting parameters to quantify underlying evolutionary relationships. Based on the proposed model, we formulate PPI network alignment as

an optimization problem and present a fast algorithm, MaWISh[2], to solve this problem [19]. Detailed experimental results show that our algorithm is able to discover conserved interaction patterns very effectively, both in terms of accuracies and computational cost. MaWISh is acknowledged as the only network alignment tool that incorporates evolutionary models into algorithms, and is viewed as the counterpart of evolutionarily motivated scoring matrices (*e.g.*, PAM and BLOSUM) in sequence alignment in the network domain [20].

In spite of algorithmic advances, development of a comprehensive infrastructure for interaction databases is in relative infancy compared to corresponding sequence analysis tools. One critical component of this infrastructure is a measure of the statistical significance of a match or a dense subcomponent. Corresponding sequence-based measures such as *E*-values are key components of sequence matching tools. In the absence of an analytical measure, conventional methods rely on computer simulations based on ad-hoc models for quantifying significance. We propose a statistical model and analysis to analytically quantify statistical significance of dense components and matches in reference model graphs [21].

We consider two reference graph models, a $G(n, p)$ model in which each pair of nodes has an identical likelihood, p, of sharing an edge, and a two-level $G(n, p)$ model, which accounts for high-degree hub nodes generally occurring in PPI networks. Experiments performed on a rich collection of PPI networks show that the proposed model provides a reliable means of evaluating statistical significance of dense patterns in these networks. We also adapt existing state-of-the-art network clustering algorithms by using statistical significance as an optimization criterion [22]. Comparison of the resulting significantly dense subgraph identification algorithm, SiDeS[3], with existing methods shows that SiDeS outperforms existing algorithms in terms of sensitivity

[2]The source code of MaWISh is publicly available at `http://www.cs.purdue.edu/homes/koyuturk/mawish`.

[3]The source code of SiDeS is publicly available at `http://www.cs.purdue.edu/homes/koyuturk/sides`.

and specificity of identified clusters with respect to available Gene Ontology (GO) [23] annotation.

The rest of this dissertation is organized as follows. In the next chapter, we provide a brief overview of molecular interaction networks, their modeling, and analysis. In Chapter 3, we present algorithms for identifying conserved subgraphs in a large collection of biological networks. In Chapter 4, we present our PPI network alignment algorithm, which is grounded in theoretical models of network evolution. In Chapter 5, we provide analytical results on statistical significance of connectivity in molecular interaction networks, and present algorithms for identification of significant dense subgraphs. We conclude our discussion with a summary and directions for future research in Chapter 6.

2. MOLECULAR INTERACTION NETWORKS

In the hierarchical organization of living organisms, cellular interactions form the bridge between individual molecules (*e.g.*, genes, mRNA, proteins, metabolites) and large-scale organization of the cell [3, 24]. Understanding these interactions provides an integrated view of the living cell, where individual molecules are dynamically orchestrated to perform cellular tasks as a system [25]. Efforts aimed at modeling, inferring, organizing, and analyzing cellular interactions have been motivated by significant advances in the understanding of genomics and have recently been the focus of considerable research attention in systems biology.

2.1 Graph Theoretic Formalisms for Biological Networks

Graph models are commonly encountered in computational analysis of cellular interactions [13, 14]. The structure of the orchestration of cellular tasks through pairwise as well as multi-way interactions between biomolecules is abstracted using various network models. Generally, in these models, molecules are represented by nodes in the network and their interactions are represented by edges (links) between these nodes. These links indicate interactions in various forms, ranging from physical binding to computationally predicted functional association, such as phylogenetic similarity. Common abstractions for molecular interactions include protein interaction networks, gene regulatory networks, metabolic pathways, and signaling pathways. While the interactions modeled using these abstractions are closely interrelated, and the underlying components of the network cannot be isolated from each other, individual models provide a simplified view of different modes of interaction, facilitating efficient organization and analysis of these interactions.

2.1.1 Protein Interaction Networks

An important class of molecular interaction data is in the form of protein-protein interactions. 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 [26]. High throughput screening methods such as two-hybrid analysis [6], mass spectrometry (MS) [5], and tandem affinity purification (TAP) [4] provide large amounts of data on the *interactome* of an increasing number of species. These data are organized into public databases, making PPI networks available for more complicated analysis tasks. Such databases include BIND [10], DIP [12], MIPS [27, 28], and MINT [29].

Experimental data may reveal either pairwise or multi-way interactions between a group of proteins, depending on the nature of screening technique. Pairwise interactions are conveniently 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 Figure 2.1(a). Multi-way interactions are either modeled with hypergraphs, in which edges are replaced by hyperedges [13], or inserted into simple graphs by contracting the multi-way interaction into a star network (spoke model) or a clique (matrix model) [30].

Protein-protein interactions can also be 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 protein-protein interactions include phylogenetic profiling [31, 32] and analysis of gene expression [33, 34], based on the premise that interacting proteins are likely to have co-evolved or be co-expressed as their cooperative task would require existence of both proteins. Since protein interaction data obtained from high-throughput screening is highly error-prone [7, 26], it is common to combine several experimental and computational sources of interaction data to obtain a reliable set of putative interactions. Such aggregated interaction networks

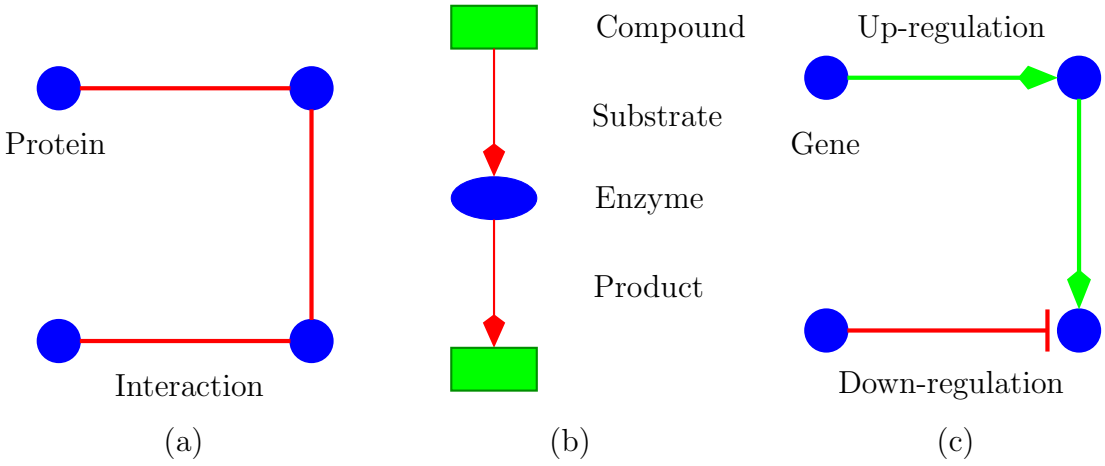


Fig. 2.1. Graph models for molecular interactions: (a) Protein interaction networks, (b) Metabolic pathways, (c) Gene regulatory networks.

are modeled using weighted graphs, where edge weights represent the likelihood of interaction between proteins and are estimated using various statistical models and techniques [8, 35–38].

2.1.2 Metabolic Pathways

Metabolic pathways comprise a historically well-studied abstraction for biological networks. They characterize the process of chemical reactions that, together, perform a particular metabolic function. With the recent progress in application of computational methods to cell biology, there have been successful attempts at modeling, synthesizing [39] and organizing metabolic pathways into public databases such as KEGG [11], MetaCyc [40], and EMP [41].

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) [42]. 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 Figure 2.1(b). It is possible to replace this model by a simpler directed graph if, for instance, we are only interested in relations 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 [43].

2.1.3 Gene Regulatory Networks

Gene regulatory networks, also referred to as genetic networks, represent regulatory interactions between pairs of genes [44]. Gene expression is regulated in various phases, including transcription [45, 46], translation [47], post-transcription, and post-translation, through various mechanisms such as mRNA degradation [48]. Regulatory interactions between genes are generally inferred from gene expression data through microarray experiments [49–51] as well as sequence analysis such as identification of regulatory motifs [52,53]. A simple and common model for gene regulatory networks is the Boolean network model. In 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 Figure 2.1(c). More complicated computational models that capture the degree of regulation through weighted graphs and/or differential equations are also used.

2.1.4 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 [54] and gene co-expression networks, which pack relations in complex expres-

sion patterns into pairwise associations between genes [34]. In the Molecule Pages database [55], 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.

2.2 Computational Analysis of Molecular Interaction Networks

Graph-theoretic modeling of biological networks provides a framework for the solution of various problems aimed at understanding modular and/or hierarchical organization of biological processes [56]. Algorithmic questions that facilitate extraction of organized and annotated information from molecular interaction networks range from simple queries to more complicated analysis tasks.

Simple queries comprise of individual and composite graph operations related to topological properties that hint on individual function as well as functional correspondence, including the following:

- *Reachability* is measured by the minimum number of interactions that separate any given pair of proteins and provides an understanding of functional association between proteins [57–59].

- *Connectivity* may be of interest for a single node (*e.g.*, degree or clustering coefficient) as well as the entire network (*e.g.*, bisection width) and provides insights on lethality of proteins and robustness of the (sub-)network [57, 59–62].

- *Density* measures the relative intensity of interactions among a given set of biomolecules and provides understanding of modularity [59, 63, 64].

More complicated analysis techniques target identification of patterns that exhibit certain interesting or unusual – hence potentially meaningful 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 [58, 63–65]. 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 [58].

- *Hierarchical decomposition* methods rely on the observation that organization of cellular processes can be modeled using hierarchical modularity [66]. These methods use hierarchical clustering algorithms for identification of functional modules [67, 68].

- *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 [45, 69–72].

- *Inferring function* of individual proteins and assigning complex memberships based on proximity, topological similarity, or other more detailed network characterists also provide a useful computational tool for extracting information from interaction data [35, 36, 73–75].

- *Impact analysis* studies the effect of alterations in specific genes, proteins, or interactions on the overall characteristics of pathways or networks and has significant importance in cancer and drug research, and engineering of cellular processes [76–78].

As more interaction data becomes available for diverse species, comparative network analysis becomes useful in extracting information from interaction networks [20]. Comparative network analysis provides understanding of conservation and divergence

of the modularity of cellular processes in an evolutionary framework for systems biology [79] and facilitates projection of functional, structural, and modular annotation for model organisms onto a diverse set of species. As in the case of sequences, key problems relating to comparative analysis of networks include aligning multiple graphs [18, 80–83], finding frequently occurring subgraphs in a collection of graphs [16, 82, 84], discovering highly conserved subgraphs in a pair of graphs, finding good matches for a subgraph in a database of graphs [85], and identification of common topological motifs [69]. In this dissertation, we emphasize on the problem of identifying conserved substructures using two different algorithmic approaches: frequent subgraph discovery and network alignment. Frequent subgraph discovery, which is discussed in Chapter 3, targets identification of unusually frequent, hence putatively conserved modular structures in a large collection of networks. Pairwise network alignment, on the other hand, compares a pair of networks to find approximate matches through evolutionary means of assessing subgraph similarity, and is discussed in Chapter 4.

An important component of computational analysis in systems biology, as well as in many other areas of molecular biology, is the assessment of the statistical significance of identified patterns. In general, for an identified pattern, it is necessary to quantify how *meaningful* the pattern is with respect to a reference model, in order to assess the biological relevance of the pattern [21, 86, 87]. Development of statistical methods to assess significance requires accurate choice of a reference model and rigorous probabilistic analysis. Such methods are quite limited for analysis of interaction networks. In Chapter 5 of this dissertation, we address this problem and present statistical models and detailed analysis for assessing the significance of connectivity and conservation in molecular interaction networks.

2.3 Relation to Other Sources of Biological Data

In addition to being inseparable from each other, molecular interactions are closely related to other sources and abstractions of biological data in various ways. In many cases, networks are inferred from other sources of data such as microarray experiments or nucleotide/aminoacid sequences. For example, while the expression of genes is modeled using a matrix in the space of genes and time, condition, or tissue, the underlying process that generates this matrix is modeled using gene regulatory networks [49], as discussed in Section 2.1.3. Furthermore, regulatory interactions, such as those involving transcriptional regulation, are commonly identified through sequence analysis and motif search [52]. Knowledge of protein structure and molecular interactions is employed mutually to derive information about each other, wherein known interactions are used to extract domains and known domain structures are used in prediction of interactions [88–90].

It is common to represent information derived from other sources of data using network models, as in the case of gene co-expression [34] networks. Such data sources are coupled with interaction data obtained from high-throughput experiments to identify modular structures such as protein complexes, functional modules, and pathways [91–93].

A reliable source of data and analysis technique that sheds light on modularity of cellular interactions is the assessment of phylogenetic relationships between protein sequences. Based on the observation that interacting proteins are likely to have co-evolved because of evolutionary constraints, the idea of predicting protein interactions and assigning functions to proteins through analysis of phylogenetic profiles has been widely employed [32, 94]. The phylogenetic profile for a protein is a vector, each entry of which signifies the existence of an ortholog in one particular genome. Hence, the basic approach in phylogenetic interaction prediction is the detection of correlated phylogenetic profiles. This idea is also extended to profiles on phylogenetic trees [95, 96] and protein family profiles [97], and enhanced through analysis of

domain profiles [31, 98] in order to capture the underlying evolutionary relationships more accurately. Recent studies reveal that the relation between co-evolution and interaction / functional association generalize from protein pairs to groups of proteins, *i.e.*, functional modules and protein complexes [99, 100]. Indeed, clustering of phylogenetic profiles for detection of functional modules is successfully applied in the analysis of prokaryotic metabolic pathways [101].

It should also be noted that combinatorial abstractions discussed in this chapter often overlook the dynamics of the cellular interactions and provide a simplified overview of the structure 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 [25].

3. IDENTIFICATION OF CONSERVED SUBGRAPHS IN BIOLOGICAL NETWORKS

In this chapter, we address the problem of finding frequently occurring molecular interaction patterns among different organisms, *i.e.*, mining a collection of biological networks for frequent subgraphs. This problem, sometimes referred to as graph mining, is particularly challenging because it relates to the NP-hard subgraph isomorphism problem. Consequently, domain-specific abstractions are necessary in order to simplify the problem. We use an abstraction based on contraction of nodes that correspond to orthologous biomolecules. We show that this simplifies the frequent subgraph discovery problem considerably, while being able to capture the underlying biological information accurately.

We devise an efficient algorithm, MULE, which is based on frequent itemset extraction to discover frequent subgraphs among these graphs taking into account the nature of molecular interaction data. Existing formulations of isomorphism based frequent subgraph extraction suffer from exponential increase in problem size due to NP-hardness of both mining and subgraph isomorphism problems. In contrast to such extant approaches, MULE avoids repeated solution of NP-hard subgraph isomorphism problem while preserving the biological relevance of identified patterns. Using the proposed algorithm, we mine protein-protein interaction networks and metabolic pathways derived from DIP, BIND, and KEGG databases. We show that MULE is able to discover biologically meaningful patterns within seconds. We also compare the computational efficiency of MULE with existing graph mining algorithms. As a stand-alone analysis technique, MULE conveys significant biological insights at rates several orders of magnitude faster than isomorphism-based graph mining algorithms. We also establish our graph simplification technique as a pruning heuristic, which may be used to discover contracted patterns to filter the data to be mined for iso-

morphic patterns. When used as a pruning heuristic, MULE reduces effective graph sizes significantly, accelerating existing techniques by several orders of magnitude.

The rest of this chapter is organized as follows. In Section 3.1, we formalize the problem, present challenges, and overview existing algorithms for frequent subgraph discovery. We present the ortholog contraction technique and establish its theoretical and biological validity in Section 3.2. In Section 3.3, we present algorithms for mining ortholog-contracted graphs. Finally, we present and discuss the interaction patterns that result from mining KEGG metabolic pathways, and DIP and BIND protein interaction networks, and illustrate the runtime characteristics of the proposed algorithm in Section 3.4.

3.1 Frequent Subgraph Discovery Problem

This study addresses the frequent subgraph discovery problem in the context of biological networks. The input to the problem is a set of graphs in which nodes correspond to biomolecules and edges correspond to interactions between these molecules. Over this set of graphs, we are looking for frequent subgraphs that are connected and isomorphic to each other. In the general setting for graph mining, isomorphism is defined with respect to the labeling of nodes. In the context of biological networks, labeling is based on the assessment of functional correspondence, as suggested by sequence homology or more comprehensive methods of functional annotation. For metabolic pathways, the hierarchical classification of enzymes provides a means for labeling nodes. In the context of protein interaction networks, proteins of different species are functionally associated through ortholog clustering. Without loss of generality, we refer to nodes as proteins, and label these nodes based on the assignment of these proteins into ortholog groups. Assessment of functional correspondence between biomolecules is discussed in detail in the next section. We do not consider edge labels (*e.g.*, compounds for metabolic pathways) for simplicity since it is relatively straightforward to extend typical graph mining algorithms to this case. We also

assume that the graphs are directed, since some molecular interactions are directed (*e.g.*, enzyme-enzyme interactions) and any undirected graph may be represented as a directed graph.

Definition 3.1.1 Interaction network. *Given a set of biomolecules V in one particular organism, a set of interactions E between these molecules, and a many-to-many mapping of these biomolecules into a set of ortholog groups $\mathcal{L} = \{l_1, l_2, ..., l_n\}$, the corresponding interaction network is modeled using a labeled graph $G = (V, E, \mathcal{L})$. Each $v \in V(G)$ is associated with a set of ortholog groups $L(v) \subseteq \mathcal{L}$. Each edge $uv \in E(G)$ represents an interaction between u and v.*

We define node labeling flexibly to allow proteins to be associated with more than one ortholog group. This is motivated by the fact that some proteins may be involved in more than one cellular process. Specifically, if domain families [102, 103] are used to relate proteins, multi-label nodes are necessary for handling multi-domain proteins. Furthermore, since observed interaction networks represent a superposition of dynamically organized interactions in spatial and temporal dimensions [76], this model accurately captures the dynamic and complex modular organization of cellular processes.

Definition 3.1.2 Subgraph of an interaction network. *A graph S is a subgraph of interaction network G, i.e., $S \sqsubseteq G$, if there is an injective mapping $\phi : V(S) \rightarrow V(G)$ such that for all $v \in V(S)$, $L(v) \subseteq L(\phi(v))$ and for all $uv \in E(S)$, $\phi(u)\phi(v) \in E(G)$.*

A subgraph S is connected if and only if for any subset $U \subset V(S)$, $\exists\ u \in U$ and $v \in V(S) \setminus U$ such that $uv \in E(S)$ or $vu \in E(S)$. In molecular interaction networks, a connected graph may be interpreted as a set of interactions related to each other through at least one molecule. Therefore, interactions that are related to a particular cellular process are expected to form a connected subgraph. Such subgraphs may also be connected to each other as a reflection of crosstalk between different processes. For

this reason, we define the frequent subgraph discovery problem as one of identifying all connected subgraphs that exist in at least an interesting number of organisms. This allows us to understand the conservation functional modules in different organisms and identify conserved links between different cellular processes.

Definition 3.1.3 Closed frequent subgraph discovery.
Input: *A set of interaction networks* $\mathcal{G} = \{G_1 = (V_1, E_1, \mathcal{L}),\ G_2 = (V_2, E_2, \mathcal{L}), ...,\ G_m = (V_m, E_m, \mathcal{L})\}$, *each belonging to a different organism, and a support threshold* σ^*.
Problem: *Let* $H(S) = \{G_i : S \sqsubseteq G_i\}$ *be the occurrence set of graph* S. *Find all connected subgraphs* S *such that* $\xi(S) = |H(S)| \geq \sigma^*|\mathcal{G}|$, *i.e.,* S *is a frequent subgraph in* $\mathcal{G}$ *and for all* $S' \sqsupset S$, $H(S) \neq H(S')$, *i.e.,* S *is closed.*

In this framework, one is interested in discovering all subgraphs that are frequent and closed. A closed subgraph is a frequent subgraph such that none of its supersets occur in the same set of organisms as itself. In other words, since the subgraphs of a pattern that occur in the same set of networks can be inferred from the larger pattern, reporting such subgraphs as frequent patterns would be redundant. Hence, by requiring the identified frequent subgraphs be closed, we ensure maximality of discovered patterns to avoid redundancy. This also allows us to identify conserved patterns for any subset of networks,taking into account the identity of each network, hence facilitating phylogenetic analysis of modularity in molecular interaction networks. This approach may also be viewed as a symmetric mining problem, where for any sufficiently large set of organisms, all maximal subgraphs that are common to the corresponding networks are of interest.

As can be inferred from the definition of a subgraph, our graph mining problem requires repeated solutions to the subgraph isomorphism problem. There exist significant literature aimed on addressing this problem and developing efficient algorithms for identifying frequent patterns in graph structured datasets [104]. Existing techniques are mostly based on syntactic approaches, providing limited performance,

which may not be adequate for many realistic applications, because of the computational challenges associated with the intractability of the problem components. Indeed, in typical applications of biological network analysis, it is necessary to run repeated queries interactively with different parameters until a satisfactory set of results is obtained. Therefore, as we elaborate in the following sections, direct application of graph mining algorithms is not feasible in the current problem setting.

3.1.1 Computational Challenges in Frequent Subgraph Discovery

Most graph mining algorithms in the literature are based on the well-studied association rule mining, or more generally, the frequent itemset discovery problem [105]. This problem can be defined as follows. Given a set of items $\mathcal{S} = \{i_1, i_2, ..., i_n\}$ and a set of transactions $\mathcal{T} = \{T_1, T_2, ..., T_m\}$ over $\mathcal{S}$, *i.e.*, $T_i \subset \mathcal{S}$ for all i, find all subsets t of $\mathcal{S}$ such that $\sigma(t) = \frac{|\{T_i \in \mathcal{T} : t \subset T_i\}|}{|\mathcal{T}|} \geq \sigma^*$. Here, $\sigma(t)$ is the support of itemset t and σ^* is the prescribed threshold on support, signifying the desired frequency of patterns to be mined. Frequent itemset mining algorithms are generally based on the lattice or downward closure property of support. This property states that an itemset cannot be frequent if even one of its subsets is not frequent [106]. Taking advantage of this property, frequent itemset mining algorithms enumerate all potentially frequent itemsets starting from smaller itemsets and effectively pruning the search space in a bottom-up fashion. In terms of frequent subgraph discovery, downward closure translates to the fact that a subgraph is frequent only if all of its subgraphs are frequent.

Most existing frequent subgraph discovery algorithms generalize state-of-the-art frequent itemset mining algorithms to structured data. However, this generalization poses significant challenges for the following reasons:

- *Subgraph isomorphism.* While counting frequencies of subgraphs in the graph database, one must verify whether a given structure is a subgraph of a graph

in the database [104]. This requires solution of the NP-complete subgraph isomorphism problem [107] at all explored points of the solution space.

- *Canonical labeling.* Frequent itemset mining algorithms dictate a lexicographic order on items and represent itemsets as ordered sets to ensure that no itemset is considered more than once. However, such an ordering of nodes and/or edges in graphs is not trivial. Furthermore, computing canonical labels for graphs in order to sort them in a unique and deterministic manner is equivalent to testing isomorphism between graphs [108]. Therefore, graph mining algorithms generally aim to minimize redundancy caused by duplicate consideration of subgraphs [109].

- *Connectivity.* While taking advantage of the downward closure property in frequent itemset mining, candidate itemsets are generated in a bottom-up fashion by extending itemsets with addition of items one by one. In the case of graph mining, extension of subgraphs is not trivial since it is necessary to maintain connectivity of candidate subgraphs, since the target frequent patterns are desired to be connected, in general.

3.1.2 Existing Algorithms for Frequent Subgraph Discovery

One of the earliest frequent subgraph discovery algorithms, Subdue [110], is based on recursively finding a subgraph that provides the best compression based on the Minimum Description Length (MDL) principle. At each step of the algorithm, the subgraph that provides maximal compression, hence is most frequent, is discovered via a beam search heuristic and replaced by a single node. This mining process is carried on recursively. In contrast to this greedy algorithm, other existing graph mining algorithms are aimed at discovering all frequent subgraphs, searching the entire space of subgraphs.

AGM [111] adapts the well-known a-priori algorithm for frequent itemset mining [106] to the identification of vertex sets that induce frequent subgraphs in a graph

database. The main feature of this algorithm is that it provides a canonical labeling for graphs based on an adjacency-matrix representation. This might be computationally infeasible for applications involving large graphs, as is the case for biological networks. FSG [108], on the other hand, provides a canonical representation based on sparse adjacency list data structure and adopts a breadth-first enumeration algorithm for discovering frequent subgraphs. Other graph mining techniques are aimed at improving these algorithms by developing more efficient canonical representations that reduce redundancy in candidate generation along with several optimization techniques to help prune the search space more efficiently.

gSpan [112] reduces the overhead introduced by the problems discussed in the previous section through a DFS-based canonical representation of graphs and enumerates the search space in a depth-first manner to achieve significant speed-up over earlier algorithms. CloseGraph [109] is an extension of gSpan designed to discover only those subgraphs that do not have a supergraph of same support to avoid redundancy in the output. FFSM [113] improves upon gSpan by reducing redundant candidate generation through a vertical search scheme based on an algebraic graph framework. SPIN [114], further speeds up frequent subgraph discovery by splitting the process into two independent tasks of mining subtrees and extending these subtrees to frequent subgraphs. This is based on the observation that major problems in graph mining are caused by the existence of cycles and a majority of these problems can be handled efficiently by avoiding cycles. GASTON [115] relies on the same idea to generate frequent substructures hierarchically by starting from paths, extending frequent paths to trees, and further extending frequent trees to graphs.

Ghazizadeh and Chawate [116] present an alternate approach for pruning the search space using summaries. In this method, graphs are summarized by superposing identically-labeled nodes and assigning weights to edges based on this superposition. Observing that the edges of a frequent subgraph must have weights greater than the frequency threshold, it is possible to prune out many subgraphs immediately by simply evaluating the weights of the edges. Our approach in this paper also relies on the

idea of contracting identically-labeled nodes, however, our algorithm is particularly designed for biological networks, in which labeling of nodes does not necessarily induce a disjoint categorization. Appropriate labeling of nodes and subsequent contraction allows us to completely avoid the subgraph isomorphism problem, while preserving the underlying biological information. Furthermore, in the analysis of biological networks, the database consists of several large graphs, while most of the existing graph mining algorithms are devised for either a large number of smaller graphs [108,112] or a single large graph [116].

The underlying source of the subgraph isomorphism problem in frequent subgraph discovery in labeled graphs is the repetition of node labels. Since there exist many proteins in an organism that are homologous to each other, this problem emerges in the analysis of biological networks as well. Since most of the existing algorithms discussed above are based on exhaustive enumeration of the search space, they are not scalable to the analysis of biological networks, which contain thousands of nodes and ten thousands of edges. However, as we shall show in the next section, if all orthologous nodes are contracted into a single node, the problem can be considerably simplified while the underlying biological information is preserved.

3.2 Algorithmic Insight: Ortholog Contraction

We propose an alternate setting for graph mining based on contraction of orthologous nodes. While simplifying the graph mining problem significantly, ortholog contraction maintains not only the correctness by preserving the underlying frequent subgraphs in the graph database, but also the biological relevance and interpretability of the discovered patterns. Here, we show the fact that the underlying frequent subgraphs in the database are preserved by ortholog contraction. Therefore, there is no loss of information resulting from our ortholog contraction technique.

Definition 3.2.1 Ortholog-contracted graph. *Given interaction network* $G = (V, E, \mathcal{L})$ *the ortholog-contracted representation of* G, $\Upsilon(G) = \bar{G} = (\bar{V}, \bar{E}, \mathcal{L})$ *is con-*

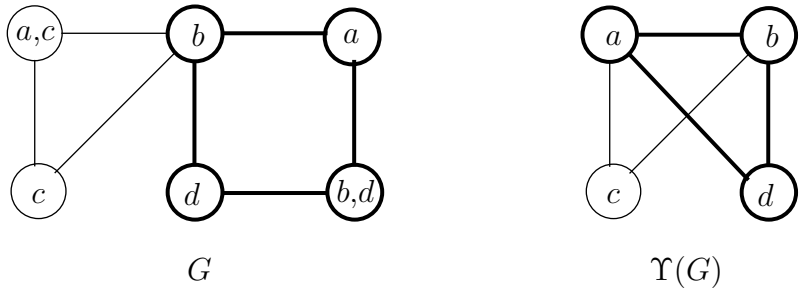


Fig. 3.1. A molecular interaction network and its ortholog-contracted representation.

structed as follows. For $1 \leq i \leq |\mathcal{L}|$*, there exists unique* $\bar{v} \in \bar{V}$ *such that* $L(\bar{v}) = \{l_i\}$*. For each* $uv \in E$ *and for all* $l_i \in L(u)$*,* $l_j \in L(v)$*, there exists* $\bar{u}\bar{v} \in \bar{E}$ *such that* $L(\bar{u}) = \{l_i\}$ *and* $L(\bar{v}) = \{l_j\}$*.*

A sample interaction network and its ortholog-contracted representation are shown in Figure 3.1. Observe that the ortholog-contracted graph of an interaction network is unique while the reverse is not necessarily true. However, all subgraphs of an interaction network are preserved in its ortholog-contracted representation, as the ortholog-contracted representations of all subgraphs of G are subgraphs of $\bar{G}$, as stated in the following theorem.

Theorem 3.2.1 Preservation of subgraphs. *Given interaction network* G*, let* $\bar{G}$ *be its ortholog-contracted representation. Then for any* $S \sqsubseteq G$*,* $\Upsilon(S) \sqsubseteq \bar{G}$*.*

Proof. Take any $S \sqsubseteq G$. Let $\bar{S} = \Upsilon(S)$ and ϕ be the appropriate mapping from $V(S)$ to $V(G)$. For each $v \in V(S)$ and $l_i \in L(v)$, there exists a unique $\bar{v} \in V(\bar{S})$ such that $L(\bar{v}) = \{l_i\}$. Since $L(v) \subseteq L(\phi(v))$, $l_i \in L(\phi(v))$. Therefore, there also exists a unique $\overline{\phi(v)} \in V(\bar{G})$ such that $L(\overline{\phi(v)}) = \{l_i\}$. Then, there is a unique injective mapping $\bar{\phi} : V(\bar{S}) \rightarrow V(\bar{G})$, where $\bar{\phi}(\bar{v}) = \overline{\phi(v)}$ for any $v \in V(S)$. Hence, for any $\bar{u}\bar{v} \in E(\bar{S})$ that results from $uv \in E(S)$, since $\exists\ \phi(u)\phi(v) \in E(G)$, there exists $\bar{\phi}(\bar{u})\bar{\phi}(\bar{v}) = \overline{\phi(u)}\ \overline{\phi(v)} \in E(\bar{G})$. Therefore, $\bar{S} \sqsubseteq \bar{G}$. □

In Figure 3.1, the ortholog-contracted representation of the bold subgraph of G is also shown in bold in $\Upsilon(G)$.

Corollary 1 Preservation of frequent subgraphs. *For a set of interaction networks* $\mathcal{G} = \{G_1, G_2, ..., G_m\}$, *let* $\bar{\mathcal{G}} = \{\Upsilon(G_1), \Upsilon(G_2), ..., \Upsilon(G_m)\}$ *be the corresponding set of ortholog-contracted graphs. If* S *is a frequent subgraph in* $\mathcal{G}$, *then* $\Upsilon(S)$ *is a frequent subgraph in* $\bar{\mathcal{G}}$.

We can interpret this result as follows. If we mine the set of ortholog-contracted graphs instead of the original set of interaction networks, we will discover a *superset* of the frequent subgraphs of the original set. In other words, *we do not miss* any frequent patterns that exist in the dataset. Therefore, it is always possible to recover the actual frequent subgraphs from the set of frequent ortholog-contracted subgraphs using an isomorphism-based graph mining algorithm. This is significantly more efficient than running the isomorphism-based algorithm on the original dataset, since mining the ortholog-contracted graph prunes out most of the infrequent substructures, thus the resulting set is significantly smaller both in terms of graph size and number of graphs. Furthermore, the idea of ortholog-contraction does not conflict with the purpose of mining molecular interaction data; as we shall show, it is very useful by itself. We elaborate on this point in the context of protein interaction networks and metabolic pathways.

3.2.1 Ortholog Contraction in Protein Interaction Networks

Recent studies on the evolution of protein interaction networks suggest that orthologous proteins that result from recent duplications are likely to share common interactions [117]. In other words, conservation of interactions between orthologous proteins translates into conservation of function. Therefore, while mining protein interaction networks for common network patterns among different species, proteins in different organisms must be related to each other through orthology.

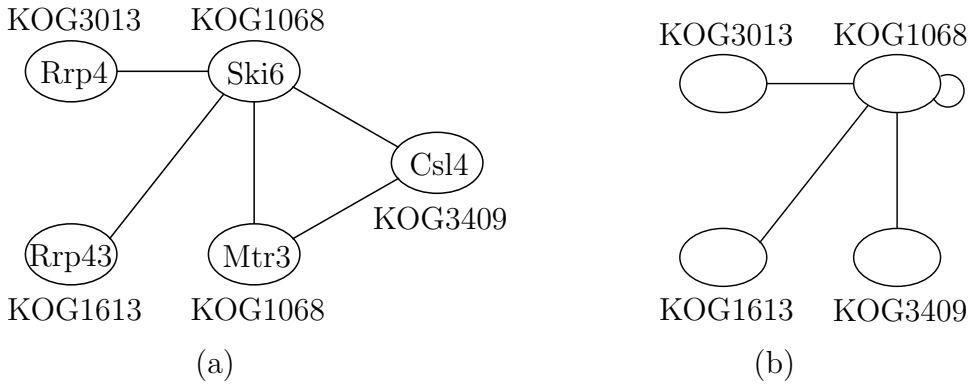


Fig. 3.2. Ortholog contraction in protein interaction networks: (a) A portion of yeast PPI network, (b) network of ortholog groups that results from contraction of proteins in the same COG cluster.

Since proteins that are evolutionarily or functionally related show significant sequence homology, a reasonable way of detecting protein families relies on sequence clustering [118,119]. A problem with inter-species protein sequence clustering is that out-paralogs, which do not possess any significant functional or evolutionary relationship since they predate the split of species, are also clustered together along with orthologs and in-paralogs [120].

Recently, ortholog families have been identified through more comprehensive *in-silico* analysis and organized into several databases, such as COG [121] and Homologene [122]. There has been relevant efforts to comprehensively identify domain families as well, including PFAM [102] and ADDA [103]. Such databases provide a reliable basis for labeling nodes in PPI networks. However, while relating nodes through domain families, interacting domains should be considered in order to avoid over-populating the contracted network.

Node contraction in protein interaction networks reduces interactions between proteins into those between ortholog groups. This is illustrated in Figure 3.2. A 5-node portion of *S. Cerevisiae* protein interaction network is shown in Figure 3.2 (a). In this figure, the common names of each protein are shown in the oval representing that protein. The nodes are labeled by their COG clusters. As a result of ortholog con-

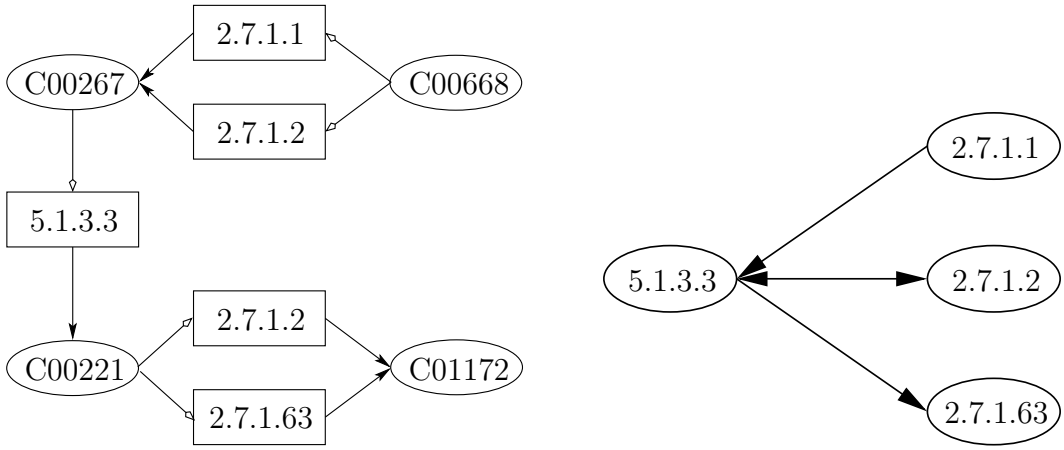


Fig. 3.3. Ortholog contraction in metabolic pathways: (a) A portion of glycolysis pathway, (b) a network of pairwise enzyme interactions that results from ortholog contraction based on enzyme nomenclature.

traction, 3'5' exoribonuclease (Mtr3) and 3'5' phosphorolytic exoribonuclease (Ski6), which belong to the same COG cluster, are contracted into single node, as shown in Figure 3.2 (b). Therefore, the interaction of these proteins with Csl4 is represented as a single interaction between ortholog groups KOG1068 and KOG3409.

3.2.2 Ortholog Contraction in Metabolic Pathways

In the directed graph model for metabolic pathways, node labels correspond to enzymes that catalyze the respective reactions. Although the biochemical properties of enzymes differ from organism to organism, enzymes are classified based on metabolic functions and protein orthology. Currently, there exists a comprehensive enzyme nomenclature that provides hierarchical classification of enzymes based on biochemical function [123]. In this enzyme nomenclature system, each enzyme is identified by its Enzyme Commission (EC) number. The numbers in the squares that represent reactions in Figure 3.3 are the EC numbers of the enzymes that catalyze these reactions.

An enzyme may catalyze multiple reactions in a particular pathway. Therefore, an enzyme class may be attached to more than one node in the corresponding graph model. However, since the edges in the directed graph model signify the producer-consumer relation between two enzymes, contracting nodes corresponding to the orthologous enzymes (*i.e.*, enzymes that belong to the same class) preserves this information [16].

The ortholog-contracted representation of the metabolic pathway graph of Figure 3.3 (a) is shown in Figure 3.3 (b). In this representation, although the node that corresponds to enzyme EC:2.7.1.2 is contracted, we do not lose the information that this enzyme not only consumes the product of EC:5.1.3.3, but also produces a compound that is consumed by the same enzyme. The only information that is hidden by this model is the fact that these two interactions between this pair of enzymes are derived from two successive reactions, which may be extracted by post-processing, as shown in the previous section.

3.3 Discovering Frequent Edgesets in Ortholog-Contracted Graphs

Once we contract orthologs into a single node for each graph, the frequent subgraph discovery problem is reduced to a generalized form of frequent itemset mining. We elaborate on this point in the following lemma.

Lemma 1 Equivalence of ortholog-contracted graphs to edge sets. *For ortholog contracted graph $\bar{G}$, define edge set $\tilde{E}(\bar{G}) = \{(l_i, l_j) : \exists uv \in E(\bar{G})$ such that $L(u) = \{l_i\}, L(v) = \{l_j\}\}$. If $\bar{S}$ is also an ortholog-contracted graph, then $\bar{S} \sqsubseteq \bar{G}$ if and only if $\tilde{E}(\bar{S}) \subseteq \tilde{E}(\bar{G})$.*

Proof. It is straightforward to see that if $\bar{S} \sqsubseteq \bar{G}$, then $\tilde{E}(\bar{S}) \subseteq \tilde{E}(\bar{G})$. Now assume that $\tilde{E}(\bar{S}) \subseteq \tilde{E}(\bar{G})$. For any $(l_i, l_j) \in \tilde{E}(\bar{S})$, there exist unique $u, v \in V(\bar{S})$ such that $L(u) = \{l_i\}$, $L(v) = \{l_j\}$, and $uv \in E(\bar{S})$. Furthermore, $(l_i, l_j) \in \tilde{E}(\bar{G})$. Therefore, there exist unique $u', v' \in V(\bar{G})$ such that $L(u') = \{l_i\}$, $L(v') = \{l_j\}$, and $u'v' \in E(\bar{G})$. Letting $\phi(u) = u'$ and $\phi(v) = v'$, we have $\bar{S} \sqsubseteq \bar{G}$. □

We can generalize this lemma to conclude that an ortholog-contracted graph is uniquely determined by the set of its edges. Therefore, mining frequent subgraphs in a collection of ortholog-contracted graphs is equivalent to mining frequent edgesets in a collection of graphs that are uniquely determined by the set of their edges. Since we are interested only in connected subgraphs, we define an edgeset to be the set of label pairs that correspond to the edges of a connected graph.

Definition 3.3.1 Edgeset. *Given a set of ortholog labels* $\mathcal{L} = \{l_1, l_2, ..., l_n\}$*, an edgeset* $F = \{e_1, e_2, ..., e_k\}$ *is a set of ordered pairs* $e_i = \{l_s, l_t\}$*, where for any subset* $F' \subset F$*, there exists* $e_i \in F'$*,* $e_j \in F \setminus F'$ *such that* $e_i \cap e_j \neq \emptyset$*.*

Definition 3.3.2 Closed frequent edgeset discovery.
Input: *Set of ortholog contracted graphs* $\bar{\mathcal{G}} = \{\bar{G}_1, \bar{G}_2, ..., \bar{G}_m\}$ *and a support threshold* σ^**.*
Problem: *For edgeset* F*, let* $H(F) = \{\bar{G}_i : F \subseteq \tilde{E}(\bar{G}_i)\}$ *be the occurrence set of* F*. Find all closed edgesets* F *that are frequent in* $\bar{\mathcal{G}}$*, i.e.,* $\xi(F) = |H(F)| \geq \sigma^* |\bar{\mathcal{G}}|$ *and for all* $F' \supset F$*,* $H(F') \neq H(F)$*.*

Observe that this problem is a generalized version of the frequent itemset mining problem. Indeed, frequent itemset mining is a special case in which the underlying graph is a clique. Therefore, a simple approach to solving this problem is to remove the connectivity constraint, and find all frequent subgraphs using a frequent itemset mining algorithm. The connected components of all frequent subgraphs provide the set of all frequent connected subgraphs. However, this approach has two drawbacks. First, although it ensures that all frequent edgesets will be discovered, it does not ensure that the discovered edgesets will be closed. Second, since the number of connected subgraphs of a clique is much larger than that of a sparse graph, this relaxation will enlarge the search space significantly, degrading computational efficiency. Therefore, a specialized algorithm for this problem, which takes into account the connectivity and maximality constraints, along with the nature of data that is derived from molecular interactions is necessary.

3.3.1 Adapting Itemset Mining to Edgeset Mining

Since frequent edgeset mining problem is closely related to the frequent itemset mining problem, we base our algorithm design on existing itemset mining algorithms taking into account the specific characteristics of biological networks.

As discussed earlier, frequent itemset mining algorithms enumerate the space of possible itemsets, exploiting the downward closure property to prune out the search space. Starting from the smallest itemsets, the occurrence of each itemset in the input transaction set is counted. Smaller frequent itemsets are extended with other frequent itemsets to generate larger itemsets that are potentially frequent. Repetitions are avoided by inducing a lexicographic ordering of items.

Two major design choices for frequent itemset mining algorithms are, the order of traversal of the enumeration tree and the method for determining the support of each itemset [124]. It is possible to traverse the itemset tree in depth-first or breadth-first fashion. Breadth-first traversal, which generates the nodes of the tree level by level, is efficient in the sense that it eliminates the maximum number infrequent itemsets at each level. However, it requires a larger memory since it stores all nodes at each level of the tree. Therefore, breadth-first traversal becomes inefficient as the tree gets deeper. Depth-first traversal, on the other hand, expands a node immediately after its itemset is discovered to be frequent, keeping storage requirement to a minimum, at the expense of exploring extra itemsets [105].

There are two possible methods for computing the support of each itemset as well. One approach is the set counting method, which makes a pass over the transaction set at each node of enumeration tree to count the number of transactions that contain the corresponding itemset. This approach is memory-efficient and well-suited to breadth-first traversal. Set intersection, on the other hand, stores the identifiers of all transactions that contain each itemset and computes the intersection of identifier sets while extending an itemset. This approach minimizes the number of passes over

the transaction set at the expense of additional memory for storing the identifier sets. This method is more appropriate for depth-first traversals.

Most closed frequent itemset mining algorithms use a depth-first traversal along with set intersection, since depth-first traversal provides the opportunity of deciding whether an itemset is closed upon its expansion [125, 126]. This combination is also appropriate for the closed frequent edgeset mining problem in biological networks for the following reasons:

- *Occurrence of subgraphs.* In contrast to association rule mining, in mining biological networks, the identity of organisms that contain the particular subgraph is of interest as well as its frequency. This is because, this set of organisms provides considerable information about the conservation of pathways, modules, and complexes, evolutionary relations between species, and the functional annotation of discovered interaction patterns. Therefore, for each edgeset explored by the algorithm, it is necessary to store the identifiers of organisms that contain this edgeset.

- *Graph size vs. database size.* In biological applications, the size of the graphs is larger than the size of typical transactions in association rule mining. For instance, a protein interaction network generally contains thousands of edges. This is also true for the cardinality of identified patterns. On the other hand, while typical data mining applications involve millions of transactions, the number of biological networks to be mined is smaller. Therefore, in mining biological networks, the enumeration tree is wider and deeper, while the amount of data to be processed at each enumeration node is smaller. This makes depth-first enumeration along with set intersection feasible and memory efficient.

3.3.2 MULE: An Efficient Algorithm for Frequent Edgeset Discovery

The key difference between frequent edgeset mining and frequent itemset mining is that in the former, we are only interested in connected subgraphs. In order to generate

procedure MINEORTHOLOGCONTRACTEDGRAPHS ($\mathcal{G}$, σ^*)

▷ **Input** $\mathcal{G}$: Set of ortholog-contracted graphs

▷ **Input** σ^*: Support threshold

▷ **Output** MFS: Set of closed frequent subgraphs

1 $\xi^* \leftarrow \sigma^*|\mathcal{G}|$

2 $\mathcal{E} \leftarrow \{e = \{l_s, l_t\} : \exists\ G \in \mathcal{G}\ s.t.\ u, v \in V(G),\ uv \in E(G), L(u) = l_s, L(v) = l_t\}$

3 **for** each $e = \{l_s, l_t\} \in \mathcal{E}$ **do**

4 $\quad H(e) \leftarrow \{G \in \mathcal{G} : \exists\ u, v \in\ V(G)\ s.t.\ uv \in E(G),\ L(u) = l_s,\ L(v) = l_t\}$

5 $\mathcal{F} \leftarrow \{e \in \mathcal{E} : |H(e)| \geq \xi^*\}$

6 $MFS \leftarrow \emptyset$

7 **for** each $e_i \in \mathcal{F}$ **do**

8 $\quad N(e_i) \leftarrow \{e_j \in \mathcal{F} : e_j \cap e_i \neq \emptyset\}$

9 $\quad$ EXTENDFREQUENTEDGESET ($\mathcal{F}$, ξ^*, MFS, $\{e_i\}$, $N(e_i)$, $\{e_1, e_2, ..., e_{i-1}\}$)

10 **return** MFS

Fig. 3.4. Main procedure for mining ortholog-contracted graphs.

all connected subgraphs in the database, we perform depth-first search on the graph constructed from all frequent edges. To avoid repetitions, we induce a lexicographic order on the edges and remember previously visited edges at each enumeration node. Assume, at any stage of the algorithm, that we have a frequent edgeset of k edges, denoted by F_k. We define the candidate set C_k to be the set of edges that are connected to the edges in F_k, but have not been previously visited. The set of edges

previously visited by the depth-first enumeration algorithm is denoted by D_k. For any candidate edge $c \in C_k$, we extend F_k as follows:

$$\begin{aligned} F_{k+1} &= F_k \cup c & \qquad D_{k+1} &= D_k = D_k \cup c, \\ N(c) &= \{e \in \mathcal{F} : e \cap c \neq \emptyset\} & C_{k+1} &= (C_k \cup N(c)) \setminus D_k. \end{aligned}$$

Here, $\mathcal{F}$ denotes the set of all frequent edges in the graph database.

The resulting algorithm for MULE is shown in Figure 3.4. This algorithm makes use of a recursive subroutine to extend frequent edgesets, which is shown in Figure 3.5. The main procedure, MINEORTHOLOGCONTRACTEDGRAPHS performs preprocessing by determining the set of frequent edges in the input graph set. It then generates each portion of the frequent edgeset tree rooted at each frequent edge by calling EXTENDFREQUENTEDGESET. Upon each invocation, EXTENDFREQUENTEDGESET tries to extend the edgeset (subgraph) by all edges in the candidate set, one by one. If the extended edgeset is frequent, then the procedure is invoked again for the extended edgeset. The algorithm stops whenever an edgeset cannot be further extended. This edgeset is then recorded, if it is not subsumed by any other recorded frequent edgeset. Upon invocation, EXTENDFREQUENTEDGESET checks whether the current frequent tree is already subsumed by other closed frequent edgesets that have previously been discovered, if so, it stops the search process. This optimization helps prune out the search space in chunks. MFS is empty on first invocation of EXTENDFREQUENTEDGESET, and is input to the procedure at each subsequent invocation, wherein it is extended with newly discovered frequent subgraphs.

Consider the input graph set shown in Figure 3.6(a). These graphs have 6 edges in all, *ab*, *ac*, *bd*, *ce*, *de*, and *ea*. Figure 3.6(b) shows the frequent edgeset tree for mining subgraphs that exist in at least 3 of the input graphs. Procedure EXTENDFREQUENTEDGESET is invoked for *ab*, *ac*, *de*, and *ea*, since these are the only frequent edges. The edgeset F, candidate set C, and the set H of identifiers of graphs that contain this edgeset are shown at each node of the edgeset tree. The sets of visited edges (D) label the branches of the tree, since these sets are shared by parent and children. At any instant, set D for a node is the one at its right-most branch. On first invocation,

procedure ExtendFrequentEdgeset ($\mathcal{F}$, ξ^*, MFS, F_k, C_k, D_k)

▷ **Input** $\mathcal{F}$: Set of frequent edges
▷ **Input** ξ^*: Frequency threshold
▷ **Input, Output** MFS: Set of maximal frequent edgesets
▷ **Input** F_k: Frequent edgeset with k edges
▷ **Input** C_k: Set of candidate edges for edgeset extension
▷ **Input** D_k: Set of already visited edges

```
1  R_k ← set of all unvisited edges reachable from F_k
2  if ∃F' ∈ MFS s.t. R_k ⊆ F' and H(F_k) ⊆ H(F') then return
3  closed ← true
4  for each c ∈ C_k do
5     D_{k+1} ← D_k ← D_k ∪ {c}
6     F_{k+1} ← F_k ∪ {c}
7     H(F_{k+1}) ← H(F_k) ∩ H(c)
8     if |H(F_{k+1})| ≥ ξ* then
9        if H(F_{k+1}) = H(F_k) then closed ← false
10       C_{k+1} ← (C_k ∪ N(c)) \ D_{k+1}
11       ExtendFrequentEdgeset(F, ξ*, MFS, F_{k+1}, C_{k+1}, D_{k+1})
12 if closed then
13    if ∄F' ∈ MFS s. t. F_k ⊆ F' and H(F_k) ⊆ H(F') then MFS ← M FS ∪ F_k
```

Fig. 3.5. Recursive procedure for extending a frequent edgeset.

the algorithm starts with edgeset $\{ab\}$, whose candidate set is $N(ab) = \{ac, ea\}$ and extends it with edge ac since the edgeset $\{ab, ac\}$ is frequent. This set cannot be

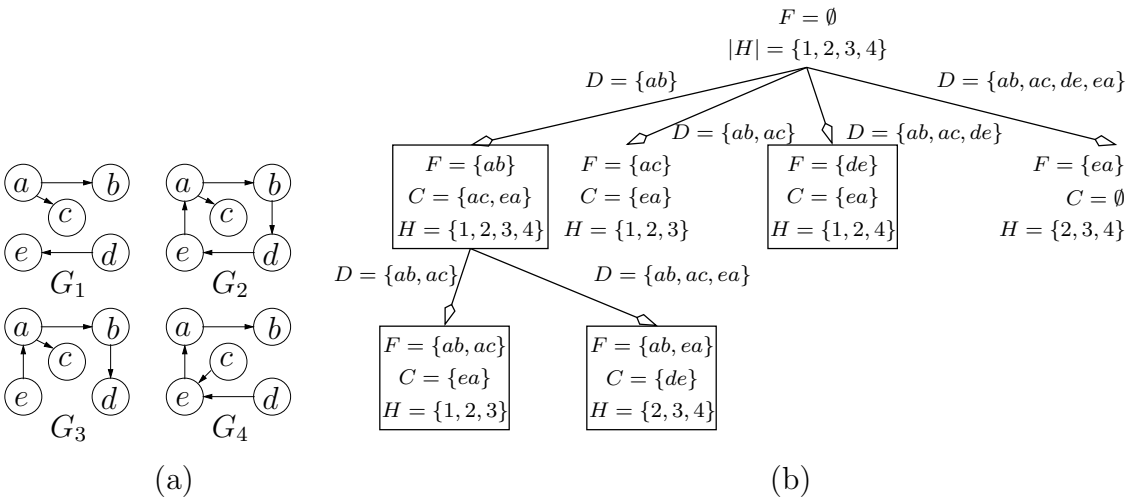


Fig. 3.6. Sample execution of MULE: (a) A collection of four ortholog-contracted graphs, (b) edgeset tree resulting from depth-first enumeration of edgesets for identification of subgraphs with frequency three.

extended by the only edge in its candidate set, ea, since the edgeset $\{ab, ac, ea\}$ is a subgraph of only two input graphs. Therefore, this edgeset is recorded as a closed frequent subgraph. Note that extension of the edgeset with edge de is not considered since this edge is not connected to the edgeset under consideration. Therefore, it never gets into the candidate edge set. Furthermore, extension of the edgeset $\{ac\}$ with edge ab is not considered since this edge has already been visited. Upon termination, the algorithm reports four closed frequent subgraphs shown in boxed nodes in the figure, which are $\{ab\}$, $\{ab, ac\}$, $\{ab, ea\}$ and $\{de\}$. Note that $\{ab\}$ is reported since its occurrence set is different from its superset $\{ab, ac\}$, hence it is closed. Although edgesets $\{ac\}$ and $\{ea\}$ are also frequent, they are not reported since they are contained in other frequent edgesets with the same occurrence set.

3.4 Experimental Results

In this section, we first present molecular interaction patterns discovered by MULE and discuss their biological interpretation. We then illustrate the runtime efficiency of

MULE, compare its execution characteristics with those of FSG and gSpan, and show that it is possible to recover actual frequent subgraphs from the contracted patterns discovered by MULE very quickly using an isomorphism-based graph mining algorithm.

3.4.1 Conserved Interaction Patterns Identified by MULE

DIP and BIND Protein Interaction Networks

We use MULE to identify conserved interaction patterns in nine eukaryotic protein interaction networks gathered from BIND [10] and DIP [12]. In order to relate the proteins in different organisms and compute ortholog-contracted graphs, we use ortholog groups derived from COG, Homologene, and sequence clustering using BLASTCLUST. We compare each homolog group in Homologene with ortholog groups in COG. If a Homologene group shares at least one protein with a COG ortholog group, we merge the Homologene group into the corresponding COG group. We then compare each protein that is not yet assigned to an ortholog group with the existing ortholog groups using BLAST. If the protein has significant sequence similarity with at least half of the proteins in a group, then we assign the protein to that ortholog group as well. For the remaining proteins, we run BLASTCLUST and create a new ortholog group from each cluster identified by BLASTCLUST. We then compute the ortholog-contracted graphs based on these ortholog groups, considering both direct and one-hop indirect interactions. The statistics of the original PPI networks and the ortholog-contracted graphs are shown in Table 3.1.

When we mine the nine PPI networks for patterns that occur in at least four of the input networks, *i.e.*, those of frequency four, we are able to identify 41 frequent connected subgraphs. The largest subgraph that is common to *H. sapiens*, *D. melanogaster*, *C. elegans*, and *S. cerevisiae* contains 18 interactions between 19 ortholog groups, which is shown in Figure 3.7(a). These interactions are associated with zinc-finger domains (KOG1721). For any combination of three organisms among

Table 3.1
Statistics of analyzed PPI networks and corresponding ortholog-contracted graphs.

Organism	PPI network		Ortholog-contracted graph		
	# proteins	#interactions	# ortholog groups	# direct interactions	# indirect interactions
A. thaliana	288	424	151	133	63
O. sativa	301	340	219	333	217
S. cerevisiae	5157	18192	1679	5327	43420
C. elegans	3345	5988	1494	2818	12968
D. melanogaster	8577	28829	2849	11088	65540
H. sapiens	4541	8577	1940	3868	23916
B. taurus	195	265	89	126	21
M. musculus	2479	2959	1213	1730	2284
R. norvegicus	696	881	445	714	761

these four, we are able to obtain larger subgraphs that are related to zinc-finger proteins. For example, *H. sapiens*, *D. melanogaster*, and *C. elegans* share 115 interactions related to zinc-finger among 83 ortholog groups, while *H. sapiens*, *D. melanogaster*, and *S. cerevisiae* share 81 interactions among 66 ortholog groups. The star shape of this interaction network is probably due to (1) numerous cellular activities that zinc-finger proteins participate in (e.g., cell division, transcription, MAP Kinase signaling, actin polymerization, and others) and (2) the large number of proteins with zinc-finger domains, both in higher and lower eukaryotes (about 1% of proteins in mammals [127]). Surprisingly, there is a significant degree of conservation of interactions among zinc-finger proteins and their partners across these diverse organisms. An interesting followup investigation would be to see how DNA binding specificities of these zinc-finger domains have evolved.

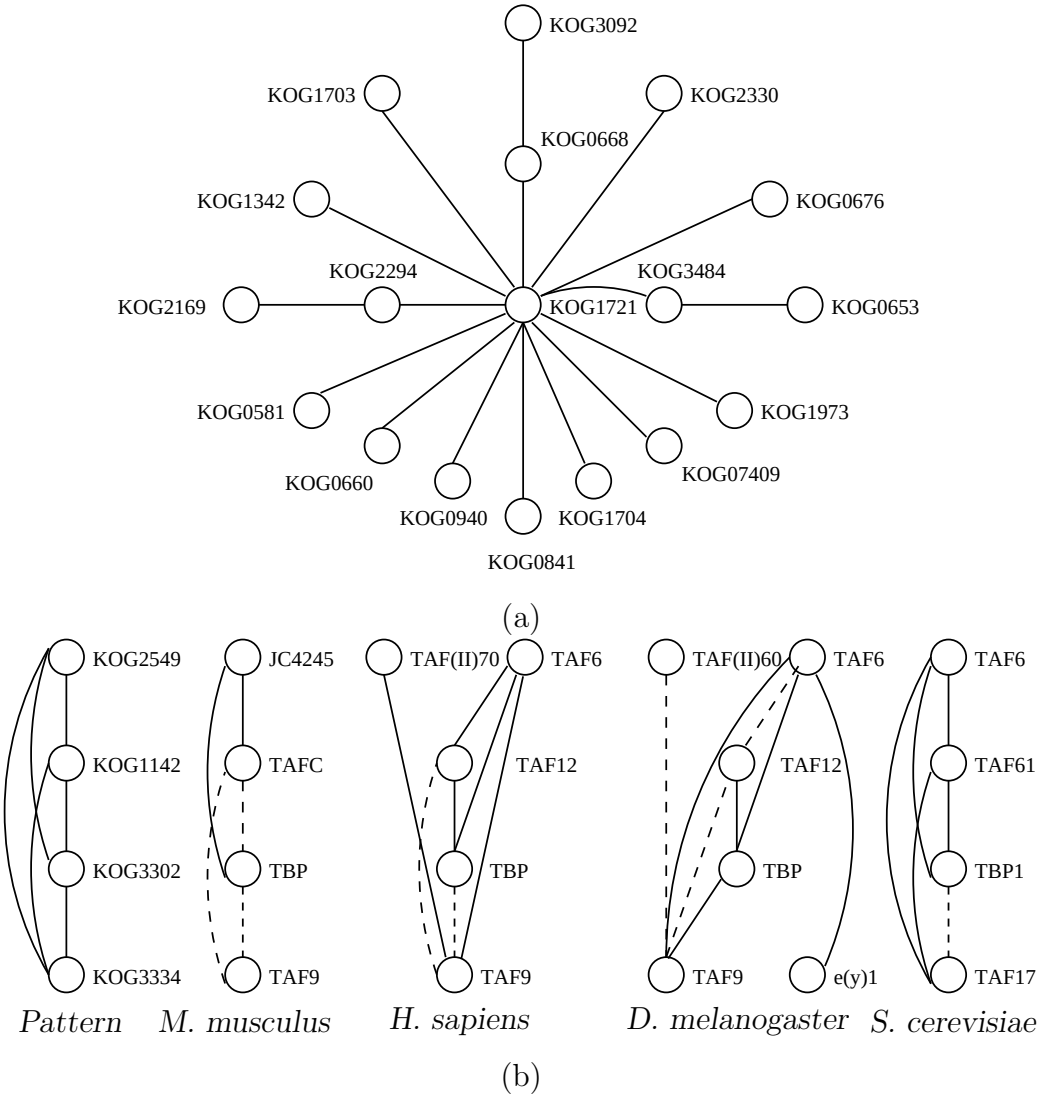


Fig. 3.7. Frequent interaction patterns that are common to four organisms.

Using the same number of organisms for the threshold, a portion of a large protein complex, TFIID, involved in transcription by RNA Polymerase II is identified as a conserved subnet in *M. musculus*, *H. sapiens*, *D. melanogaster*, and *S. cerevisiae* [128]. This conserved subnet is shown in Figure 3.7(b). The mapping of these interactions on each organism are also shown in the figure, where direct and indirect interactions are shown by solid and dashed edges, respectively. In *S. cerevisiae*, this protein com-

plex consists of one TATA-Binding Protein (TBP) and at least 14 TATA-Associated Factors (TAFs); yet in the conserved subnetwork, only 4 are found [128]. One hypothesis explaining this observation is that the TAFs present in the conserved network have greater role in promoting transcription relative to other TAFs that are absent.

When we lower the frequency threshold to 3, MULE identifies much larger number of conserved interaction patterns, specifically 158 frequent subgraphs. Four of these patterns and their mapping on the corresponding organisms are shown in Figure 3.8. Almost all proteins involved in these conserved subnets are well-annotated for *S. cerevisiae*, which facilitates mapping of these annotations to other organisms that share these interaction patterns. The subnet in Figure 3.8(a) is a pathway associated with small nuclear ribonucleoprotein complex and is conserved in *D. melanogaster*, *C. elegans*, and *S. cerevisiae*. Proteins Lsm1-7 make up a complex that participates in mRNA degradation and splicing [129]. Proteins Smx3 and Smd2 are sequence homologs of subunits in this complex. The interactions among components of Actin-related protein Arp2/3 complex, conserved in *B. taurus*, *H. sapiens*, and *S. cerevisiae*, are shown in Figure 3.8(b). This complex is involved in actin nucleation. There are 7 components known in all for this complex in *S. cerevisiae*, where Arc18 is missing in the conserved subnet [130]. In the same study, Arc40 is indicated to be essential for viability, which may explain why Arc40 has greater number of interacting partners than the other proteins present in the conserved network. In Figure 3.8(c), two endosomal sorting complexes, ESCRT-II (Vps22, Vps25, and Vps36) and ESCRT-III (Vps20, Vps24, and Vps32), are shown to be conserved together in *D. melanogaster*, *S. cerevisiae*, and *H. sapiens*. These two complexes take part in the multivesicular-body pathway and act downstream of another protein complex, ESCRT-I [131]. Finally, in Figure 3.8(d), dense interactions between a collection of proteins involved in vesicle transport are detected [132]. These interactions are conserved in *D. melanogaster*, *S. cerevisiae*, and *R. norvegicus*.

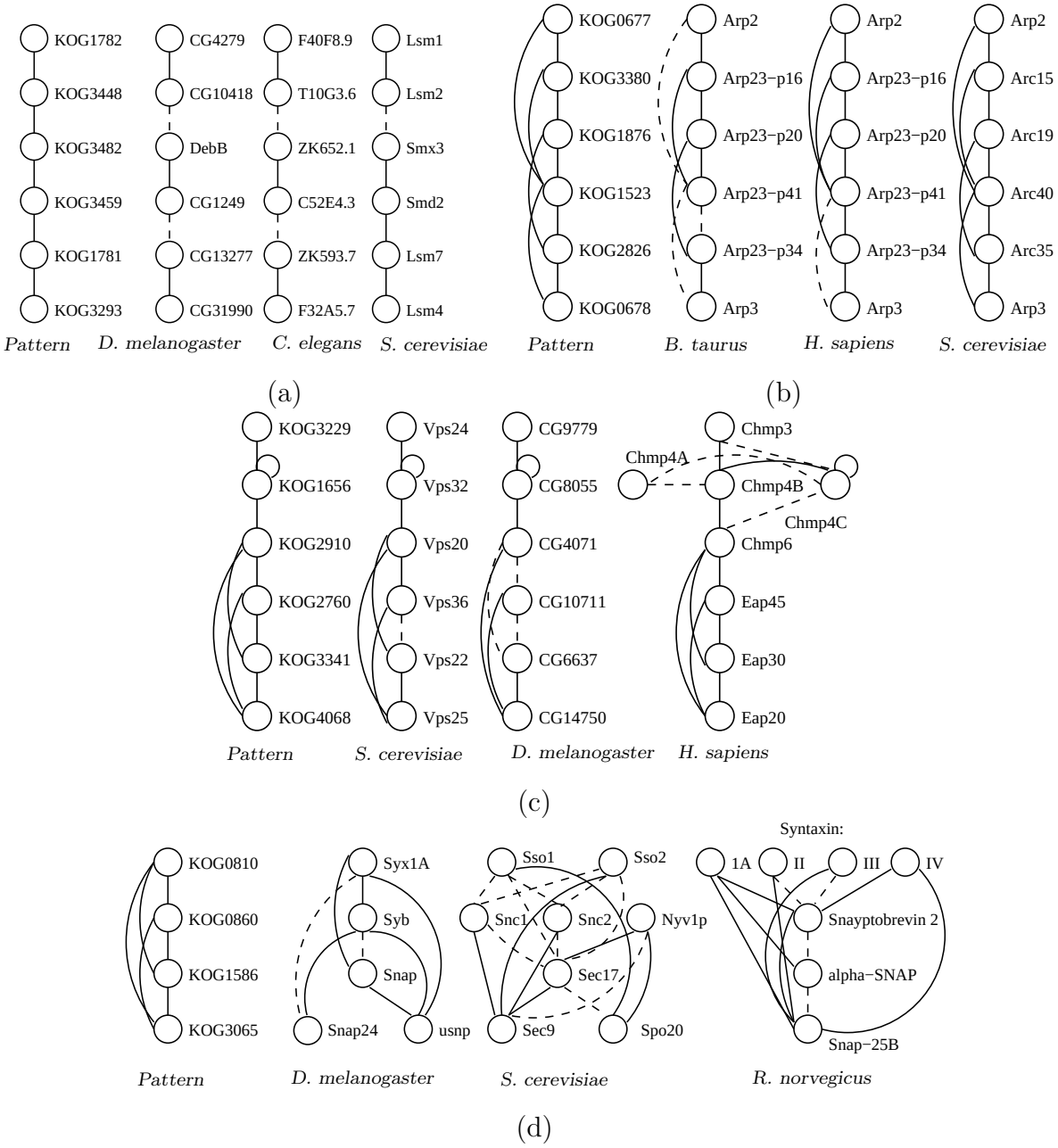


Fig. 3.8. Sample interaction patterns that are conserved in three organisms.

Frequent Sub-pathways in KEGG Metabolic Pathways

Using the proposed algorithm, we mine several pathway collections extracted from the KEGG metabolic pathway database. KEGG currently contains a large database

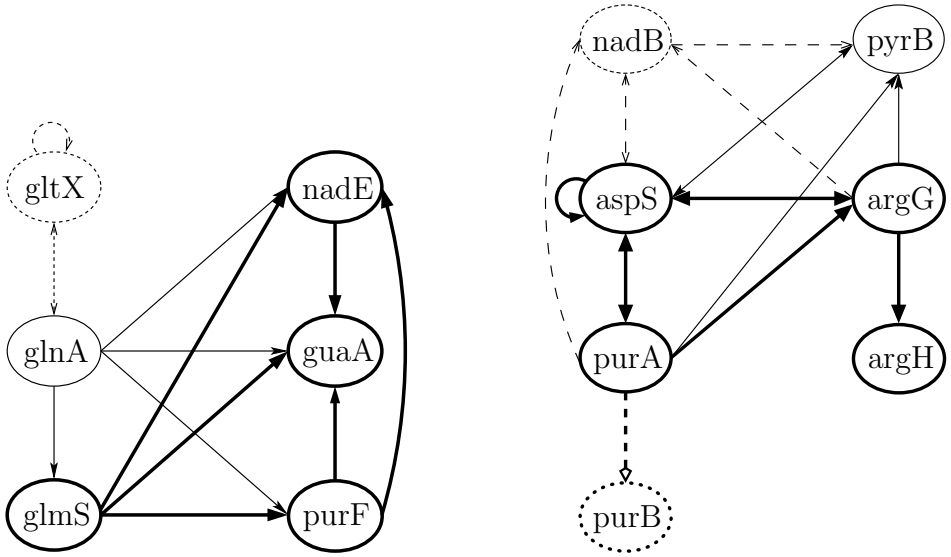


Fig. 3.9. Sample frequent subgraphs in Glutamate and Alanine-Aspartate metabolic pathways.

of pathway maps for several metabolic processes, including carbohydrate, energy, lipid, nucleotide, and aminoacid metabolism for 157 organisms. We mine several pathways belonging to different metabolisms for different organisms. Sample frequent sub-pathways discovered in pathway collections that belong to glutamate and alanine metabolisms are shown in Figure 3.9. The nodes of the displayed graphs are labeled by KEGG ID's of enzymes, which can be queried on KEGG web site for detailed information.

We are able to observe fairly large sub-pathways that are frequent. For example, a sub-pathway of glutamate metabolism that contains 4 nodes and 6 edges occurs in 45 (29%) of the 155 organisms. This sub-pathway is shown by bold nodes and edges in Figure 3.9(a). It is composed of enzymes glmS (2.6.1.16 - glucosamine-fructose-6-phosphate-aminotransferase), guaA (6.3.5.2 - GMP synthase), nadE (6.3.5.1 - NH(3)-dependent NAD(+) synthetase), and purF (amidophosphoribosyltranferase). In this sub-pathway, all enzymes are related by L-Glutamine.

Mining the pathways for different support thresholds allows evaluation of frequent sub-pathways in a multi-level fashion. For instance, when we reduce the required support threshold to 19.3% (30 organisms) for glutamate metabolism, the largest sub-pathway we are able to discover consists of 5 nodes and 10 edges, which is a supergraph of the previous one. This sub-pathway is shown in the figure by solid nodes and edges. As seen in the figure, this pathway contains enzyme glnA (6.3.1.2 - glutamine synthetase), which is also related to the other enzymes by L-glutamine. Further reducing the support threshold to 14.2% (22 organisms), we are able to discover a sub-pathway of 6 nodes and 13 edges, which is the entire graph shown in the figure. This pathway is also a supergraph of the previous one, with gltX (6.1.1.17 - glutamyl-tRNA synthetase) added, which interacts bidirectionally with glnA through L-Glutamate. The self-loop for gltX implies that this enzyme takes part in two consecutive reactions, which are part of the observed frequent sub-pathways. The original frequent sub-pathway extracted from this largest frequent ortholog-contracted subgraph is shown in Figure 3.10(a).

In Figure 3.9(b), largest of the frequent sub-pathways that are discovered in alanine-aspartate metabolism for three different levels of support threshold are shown. The bold sub-pathway of 5 nodes and 8 edges occurs in 50 (32.1%) of the 156 organisms, the solid one with 5 nodes and 11 edges occurs in 30 (19.2%) of the organisms, and the entire graph of 6 nodes and 16 edges occurs in 18 (11.5%) of the organisms. Note that enzyme purB (4.3.2.2 - adenylosuccinate lyase) and its interaction with purA (6.3.4.4 - adenylosuccinate synthetase) through adenylosuccinate (N6-(1,2-Dicarboxyethyl)-AMP), shown in dotted lines in the figure, is included in the most frequent sub-pathway of alanine-aspartate metabolism but is excluded from the larger sub-pathways of lower frequency, which is interesting to note. The original frequent sub-pathway extracted from the largest frequent ortholog-contracted subgraph is shown in Figure 3.10(b).

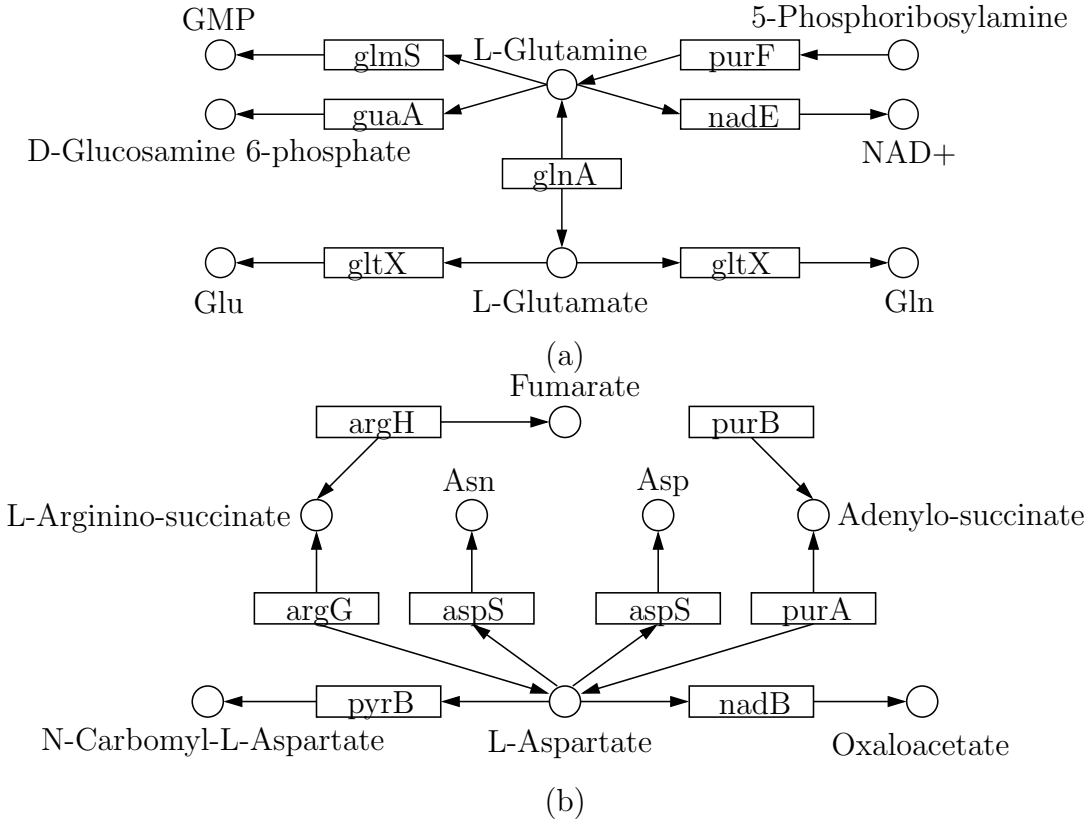


Fig. 3.10. Frequent sub-pathways of Glutamate and Alanine-Aspartate metabolism, extracted from frequent subgraphs discovered by MULE.

3.4.2 Runtime Efficiency

In this section, we compare MULE to two existing graph mining algorithms, FSG [108] and gSpan [112] to illustrate the effectiveness of node-contraction in terms of runtime performance. All experiments reported in this section are performed on a Pentium-IV 3.0 GHz workstation with 512 MB RAM.

To evaluate runtime efficiency, we compare the performance of isomorphism-based algorithms and MULE on metabolic pathways. We choose metabolic pathways as they are smaller than PPI networks, for which the isomorphism-based algorithms generally do not scale. Furthermore, metabolic pathways are available for a larger number of species, providing an appropriate setting for evaluation of scalability to a large num-

Table 3.2
Comparison of runtime performances of an isomorphism-based frequent subgraph discovery algorithm, FSG, and Mule, which is based on ortholog contraction.

		FSG			Mule		
Dataset	Min. Sup. (%)	Runtime (secs.)	Subg. size	# pats.	Runtime (secs.)	Subg. size	# pats.
Glutamate	20	0.2	9	12	0.01	9	12
	16	0.7	10	14	0.01	10	14
	12	5.1	13	39	0.10	13	39
	10	22.7	16	34	0.29	15	34
	8	138.9	16	56	0.99	15	56
Alanine	24	0.1	8	11	0.01	8	11
	20	1.5	11	15	0.02	11	15
	16	4.0	12	21	0.06	12	21
	12	112.7	17	25	1.06	16	25
	10	215.1	17	34	1.72	16	34

ber of networks. In all of our experiments, we observe that Mule runs much faster than both FSG and gSpan on the graph collections obtained from metabolic pathway datasets. First, we are not able to obtain results from gSpan on the raw directed graphs obtained directly from KEGG metabolic pathways. We suspect that gSpan is not able to respond to these queries because of memory limitations. However, as we illustrate further in this section, gSpan runs very quickly on datasets that are filtered using Mule. The performance comparison of Mule and FSG is shown in Table 3.2. The runtimes of Mule and FSG along with the number of frequent subgraphs (patterns) and the size of (number of edges in) the largest pattern are shown in the table. As is evident from the figures in the table, Mule runs much faster than FSG by several orders of magnitude. Note that FSG always returns maximal frequent sub-

Table 3.3
Extraction of contracted patterns discovered by MULE using isomorphism-based algorithms.

Glutamate metabolism, $\sigma^* = 8\%$				Alanine metabolism, $\sigma^* = 10\%$			
Size of contd. pattern	Extraction time (secs.) FSG	gSpan	Size of extd. pattern	Size of contd. pattern	Extraction time (secs.) FSG	gSpan	Size of extd. pattern
15	10.8	1.12	16	16	54.1	10.13	17
14	12.8	2.42	16	16	24.1	3.92	16
13	1.7	0.31	13	12	0.9	0.27	12
12	0.9	0.30	12	11	0.4	0.13	11
11	0.5	0.08	11	8	0.1	0.01	8
# of patterns: 56				# of patterns: 34			
Total runtime							
FSG: 138.9 secs.				FSG: 215.1 secs.			
MULE+FSG: 0.99+100.5 secs.				MULE+FSG: 1.72+160.6 secs.			
MULE+gSpan: 0.99+16.8 secs.				MULE+gSpan: 1.72+31.0 secs.			

graphs. MULE, on the other hand, sometimes returns supersets of frequent subgraphs because of contraction. In our experiments on metabolic pathways, we notice that these supersets are rare and can be easily identified upon examination. Observe that in Table 3.2, the number of frequent subgraphs discovered by FSG and MULE are the same for all support values in both datasets. This shows that the frequent patterns discovered by the two algorithms correspond to the same set of patterns, while some of these patterns are smaller in MULE, since an edge that actually appears at different locations in the subgraph is contracted into one edge by MULE.

The supersets returned by MULE can be reprocessed through FSG or gSpan and exact frequent subgraphs can be extracted very quickly. This is illustrated in Table 3.3. In the table, we display the extraction of five largest subgraphs that are

discovered by MULE for both datasets. These results show that MULE can be used in a different setup for analysis of biological networks as well. In this setup, a user first mines the graph collection of interest using MULE. Note that, since MULE is fast enough, this can be done repeatedly to tune the minimum support value to obtain the most interesting set of discovered patterns. Upon examination of frequent subgraphs discovered by MULE, the user may choose the patterns of special interest among these. Then, the actual patterns that correspond to these contracted patterns can be extracted by filtering the database and running one of the isomorphism-based graph mining algorithms such as FSG and gSpan. Filtering the graph database reduces the size of the search space substantially in terms of both number and size of graphs to be mined. Indeed, as evident from Table 3.3, the largest subgraphs that are discovered by MULE are extracted within seconds. In addition, extracting the entire set of frequent subgraphs discovered by MULE takes much less time than mining the original dataset directly, using one of the isomorphism-based algorithms without any preprocessing. As seen in the table, we are able to discover all frequent ($\sigma^* = 8\%$) subgraphs on Glutamate pathway collection in 17.8 seconds through preprocessing with MULE followed by isomorphism-based mining with gSpan. Recall that we are not able to mine the original datasets with gSpan alone. Similarly, a combination of MULE and FSG is able to mine this dataset in 101.5 seconds, while FSG alone spends 138.9 seconds to complete the same task. This improvement in runtime (factor of roughly 8) increases rapidly with database size. As databases grow, node contraction is the only known viable approach. In conclusion, while MULE is established as a fast tool for discovering frequent patterns in biological networks in a biologically interpretable fashion, it can also be used to improve other graph mining algorithms. Note also that in the case of protein interaction networks, node contraction is generally necessary for understanding evolutionary relationships.

3.4.3 Discussion

MULE is able to detect known functional modules from the interaction networks by exploiting their conservation among different organisms (Figures 3.7 and 3.8). Although our results are limited by the availability of the interaction data, it appears that the conservation of functional modules is a wide-spread phenomenon observed in numerous cellular activities. Interactions among subunits of protein complexes involved in transcription, mRNA degradation and splicing, actin nucleation, endosomal sorting, and vesicle transport are significantly conserved in yeast and higher eukaryotes, such as humans. This suggests that as more interaction data becomes available, MULE can be used to automatically map functional organization of proteins of a query organism based on the interaction networks of others.

In terms of runtime efficiency, MULE outperforms existing graph mining algorithms by demonstrating scalability to increasing network and pattern size. Furthermore, in contrast to existing multiple alignment algorithms, MULE is scalable to a very large number of organisms (networks), which makes it applicable for real-time analysis in realistic settings. Existing multiple network alignments are based on network cartesian product, *i.e.*, they create a new node for every group of homologous proteins from each species [82]. For instance, while aligning m networks, if an ortholog group contains k proteins in each species, then a cartesian-product based alignment algorithm represents this group with k^m nodes. Such exponential time complexity in terms of the number of organisms make such algorithms infeasible for a large number of networks; indeed such algorithms are applied to at most three organisms up to date. MULE, on the other hand, represents such a group of homologous proteins with only m nodes, and demonstrates scalability to hundreds of organisms. However, in contrast to multiple alignment algorithms, MULE identifies only exact matches rather than approximate matches.

An important problem in large-scale analysis of interaction networks for a growing number of networks arises from the fact that interaction data is noise laden [26].

Through the ortholog contraction approach in a graph mining based framework, MULE provides robustness and error-correction ability in two ways: (i) interactions that are conserved across a large set of networks are unlikely to arise from noise, (ii) false negatives (existing interactions missing in the data) are likely to be corrected through ortholog contraction, since if two proteins interact, proteins in the same organism that are similar to those in function and sequence are also likely to interact [117]. A limitation of MULE, however, is the modeling of interaction networks using unweighted graphs, whereas weighting interactions provides a more reliable way of accounting for noise [82].

4. ALIGNMENT OF PROTEIN INTERACTION NETWORKS

While frequent subgraph discovery algorithms provide fast identification of conserved patterns in biological networks, approximate matching algorithms that target identification of similar subgraphs that are of certain evolutionary proximity are also necessary [20]. In the case of sequences, this problem is known as the sequence alignment problem, and is one of the fundamental tools in comparative genomics [1, 133]. As is the case with sequences, pairwise and multiple alignment of graphs [18, 80–83], as well as finding good matches for a subgraph in a database of graphs [85] are important problems in comparative network analysis.

A publicly available tool, PathBLAST, adapts the ideas in sequence alignment to PPI networks to discover conserved protein pathways across species [85, 134]. By restricting the alignment to pathways, *i.e.*, linear chains of interacting proteins, this algorithm simplifies the problem, while preserving the biological implication of discovered patterns. PathBLAST accounts for gaps and mismatches by allowing non-repeated jumps and matching of non-orthologous proteins, based on the notion that the orthologous counterpart of a pair of interacting proteins in one species will, likely, be indirectly interacting in the other [134]. Similarly, Pinter et al. [135] align metabolic pathways based on subtree homeomorphism, observing that this model not only simplifies the problem by avoiding cycles, but also can describe variations in metabolic pathways effectively.

In addition to pathways and trees, a more general pattern structure is in the form of subgraphs induced by a group of proteins. Such subgraphs may provide insight into the conservation of functional modules and protein complexes, since these building blocks of cellular processes manifest themselves as dense or highly connected subgraphs in the PPI network [63, 92]. Indeed, in a recent study, Sharan et al. [81] show

that cross-species network comparison in terms of general subgraphs provides novel biological insights through incorporation of knowledge about two different networks. Specifically, they identify conserved complexes in bacteria and yeast by constructing an orthology graph with nodes that correspond to pairs of orthologous proteins, one from each species. The edges of the orthology graph are weighted according to a probabilistic framework that compares null and conserved complex models based on log-likelihood, which takes into account the conservation and density of interactions. The idea of constructing product graphs by joining orthologous nodes is also applied to the comparative analysis of PPI networks that belong to multiple species [82].

Based on the understanding of the structure of PPI networks that are available for several species, theoretical models that focus on understanding the evolution of protein interactions are developed [136–140]. Among these, the duplication/divergence model is shown to be promising in explaining the power-law nature of PPI networks [139]. In this paper, we propose a framework for alignment of PPI networks based on these evolutionary models. As in [81, 82, 85, 134], we construct product graphs by matching pairs of orthologous nodes. In contrast to these studies, however, our framework is based on concepts of matches, mismatches, and duplications, and edges are weighted in order to reward or penalize these evolutionary events. This can be viewed as an extension of the concept of alignment in the sequence domain to that in network domain. Hence, our model provides a general framework that allows selection of parameters based on existing information about the conservation and divergence of proteins and their interactions, which can be refined in the light of a diverse range of mathematical models for network evolution [62, 138, 141–144]. According to a recent survey of comparative network analysis [20], on the timeline of network alignment algorithms, our method corresponds to similarity matrices (*e.g.*, PAM, BLOSUM), which had a significant impact on sequence alignment methods by incorporating evolutionary information in sequence alignment. Indeed, our method is pointed out to be the sole network alignment tool that incorporates evolutionary models into its algorithms.

We reduce the alignment problem to a graph-theoretic optimization problem and propose efficient heuristics to solve this problem. Experimental results based on an implementation of our framework show that the proposed algorithm is able to discover conserved interaction patterns very effectively. The proposed algorithm, MAWISH, can also be straightforwardly adapted to finding matches for a subnet query in a database of PPI networks.

The rest of this chapter is organized as follows: we start with a brief overview of duplication/divergence models for the evolution of PPI networks in Section 4.1. In Section 4.2, we define the alignment problem based on these models of evolution. We then formulate the problem as a graph optimization problem in Section 4.3, and propose efficient heuristics for the solution of the problem in Section 4.4. In Section 4.5, we discuss possible extensions for the proposed model. Finally, we illustrate the effectiveness of the proposed framework on comprehensive pairwise alignment of the PPI networks for three eukaryotic species in Section 4.6.

4.1 Theoretical Models for Evolution of PPI Networks

There exist a number of studies aimed at understanding the general structure of PPI networks. These studies suggest that PPI networks can generally be modeled using power-law graphs, *i.e.*, the relative frequency of proteins that interact with k proteins is roughly proportional to $k^{-\gamma}$, where γ is a network-specific parameter [60]. In order to explain this power-law nature, Barábasi and Albert propose [60] a network growth model based on preferential attachment, which is able to generate networks with degree distribution similar to PPI networks. According to this model, networks expand continuously by addition of new nodes and these new nodes prefer to attach to well-connected nodes when joining the network. Observing that older proteins are better connected, Eisenberg and Levanon [137] explain the evolutionary mechanisms behind such preference by the strength of selective pressure on maintaining connectivity of strongly connected proteins and creating proteins to interact with

them. Furthermore, in a relevant study, it is observed that the interactions between groups of proteins that are temporally close in the course of evolution are likely to be conserved, suggesting synergistic selection during network evolution [144].

A common model of network evolution that explains preferential attachment is the duplication/divergence model, which is based on gene duplications [117,138–140]. 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. An example of protein duplication is shown in Figure 4.1. 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 non-interacting proteins, caused by mutations that change protein surfaces. Examples of elimination and emergence of interactions are also illustrated in Figure 4.1. In the example shown in this figure, starting with three interactions between three proteins, protein u_1 is duplicated to add u_1' into the network together with its interactions (dashed circle and lines). Then, u_1 loses its interaction with u_3 (dotted line). Finally, an interaction between u_1 and u_1' is added to the network (dashed line). If an elimination or emergence is related to a recently duplicated protein, it is said to be correlated; otherwise, it is uncorrelated [138]. Since newly duplicated proteins are more tolerant to interaction loss because of redundancy, correlated elimination is generally more probable than emergence and uncorrelated elimination [139]. It is also theoretically shown that network growth models based on node duplications generate power-law distributions [136].

Since the elimination of interactions is related to sequence-level mutations, one can expect a positive correlation between similarity of interaction profiles and sequence similarity for paralogous proteins [117]. Indeed, the interaction profiles of duplicated proteins tend to almost totally diverge in about 200 million years, as

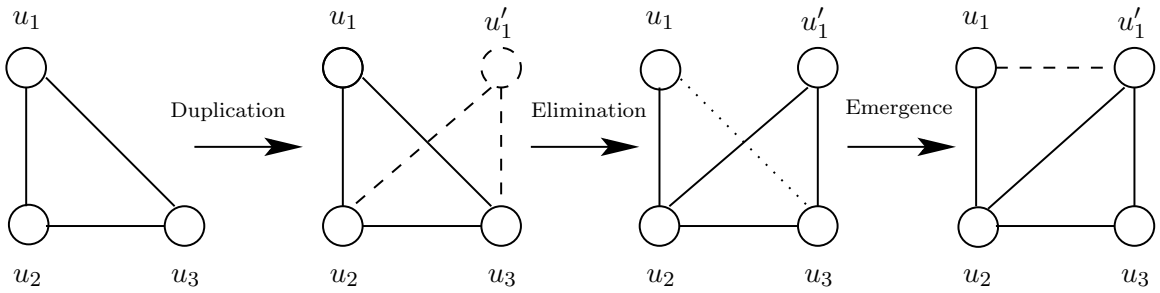


Fig. 4.1. Duplication/divergence model for evolution of PPI networks.

estimated on the yeast interactome. On the other hand, the correlation between interaction profiles of duplicated proteins is significant for up to 150 million years after duplication, with more than half of interactions being conserved for proteins that are duplicated less than 50 million years back [117]. Consequently, when we consider the PPI networks that belong to two separate species, the in-paralogs will be likely to have more common interactions than out-paralogs. Here, we use the terms in-paralog and out-paralog for proteins that are duplicated before and after speciation, respectively. While comparatively analyzing the proteome and interactome, it is important to distinguish in-paralogs from out-paralogs since the former are more likely to be functionally related. This, however, is a difficult task since out-paralogs also show sequence similarity.

In order to accurately identify and interpret conservation of interactions, complexes, and modules across species, we base our framework for the local alignment of PPI networks on duplication/divergence models. While searching for highly conserved groups of interactions, we evaluate mismatched interactions and paralogous proteins in light of the duplication/divergence model. Introducing the concepts of match (conservation), mismatch (emergence or elimination) and duplication, which are in accordance with widely accepted models of evolution, we are able to discover alignments that also allow speculation about the structure of the network in the common ancestor.

4.2 PPI Network Alignment Problem

A PPI network is conveniently modeled using an undirected graph $G(U, E)$, where U denotes the set of proteins and $uu' \in E$ denotes an interaction between proteins $u \in U$ and $u' \in U$. For pairwise alignment of PPI networks, we are given two PPI networks belonging to two different species, denoted by $G(U, E)$ and $H(V, F)$. The homology between a pair of proteins is quantified by a similarity measure that is defined as a function $S : (U \cup V) \times (U \cup V) \rightarrow \Re$. For any $u, v \in U \cup V$, $S(u, v)$ measures the degree of confidence in u and v being orthologous, where $0 \leq S(u, v) \leq 1$. If u and v belong to the same species, then $S(u, v)$ quantifies the likelihood that the two proteins are in-paralogs. S is expected to be sparse, *i.e.*, each protein is expected to have only a few potential orthologs. We discuss the methodology for deriving similarity scores from sequence alignments in Section 4.2.3.

For PPI networks $G(U, E)$ and $H(V, F)$, a *protein subset pair* $P = \{\tilde{U}, \tilde{V}\}$ is defined as a pair of protein subsets $\tilde{U} \subseteq U$ and $\tilde{V} \subseteq V$. Any protein subset pair P induces a local alignment $\mathcal{A}(G, H, S, P) = \{\mathcal{M}, \mathcal{N}, \mathcal{D}\}$ of G and H with respect to S, characterized by a set of duplications $\mathcal{D}$, a set of matches $\mathcal{M}$, and a set of mismatches $\mathcal{N}$. 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 split, or an experimental error. Thus, mismatches are penalized to account for the divergence from the common ancestor. We provide formal definitions for these three concepts to construct a basis for the formulation of local alignment as an optimization

problem. Note that although PPI networks are undirected graphs, interactions are regarded as ordered pairs in the following definitions for convenience, *i.e.*, for an interaction $uu' \in E$, there is also an interaction $u'u \in E$, which is essentially the same interaction.

Definition 4.2.1 Local Alignment of PPI networks.
Given protein interaction networks $G(U, E)$, $H(V, F)$, let functions $\Delta_G(u, u')$ and $\Delta_H(v, v')$ denote the distance between two corresponding proteins in the interaction graphs G and H, respectively. Given a pairwise similarity function S defined over the union of their protein sets $U \cup V$, and a distance cutoff $\bar{\Delta}$, any protein subset pair $P = (\tilde{U}, \tilde{V})$ induces a local alignment $\mathcal{A}(G, V, S, P) = \{\mathcal{M}, \mathcal{N}, \mathcal{D}\}$, where

$$\begin{aligned}\mathcal{M} = \ \{ \ & u, u' \in \tilde{U}, v, v' \in \tilde{V} : S(u, v) > 0, \ S(u', v') > 0, \\ & ((uu' \in E \wedge \Delta_H(v, v') \leq \bar{\Delta}) \vee (vv' \in F \wedge \Delta_G(u, u') \leq \bar{\Delta})) \ \ \}\end{aligned} \tag{4.1}$$

$$\begin{aligned}\mathcal{N} = \ \{ \ & u, u' \in \tilde{U}, v, v' \in \tilde{V} : S(u, v) > 0, \ S(u', v') > 0, \\ & ((uu' \in E \wedge \Delta_H(v, v') > \bar{\Delta}) \vee (vv' \in F \wedge \Delta_G(u, u') > \bar{\Delta})) \ \ \}\end{aligned} \tag{4.2}$$

$$\mathcal{D} = \ \{ \ u, u' \in \tilde{U} : S(u, u') > 0 \ \} \ \cup \ \{ \ v, v' \in \tilde{V} : S(v, v') > 0 \ \} \tag{4.3}$$

Each match $M \in \mathcal{M}$, mismatch $N \in \mathcal{N}$, and duplication $D \in \mathcal{D}$ are associated with scores $\mu(M)$, $\nu(N)$ and $\delta(D)$, respectively.

Following the definition of match and mismatch, while assessing the conservation of interactions, we take into account not only direct but also indirect interactions. If two proteins directly interact with each other in one organism, and their orthologs are reachable from each other via at most $\bar{\Delta}$ interactions in the other, we consider this a match. Conversely, a mismatch corresponds to the situation in which two proteins cannot reach each other via $\bar{\Delta}$ interactions in one network while their orthologs directly interact in the other. This approach is motivated by two observations. First, proteins that are linked by a short alternate path are more likely to tolerate losing their interaction because of relaxation of evolutionary pressure. Second, high-throughput methods such as TAP [4] identify complexes that are associated with a single central

protein and these complexes may be recorded in the interaction database as star networks with the central protein serving as a hub. Therefore, all proteins that are part of a particular complex can be viewed as interacting by setting $\bar{\Delta} = 2$.

4.2.1 Scoring Match, Mismatch, and Duplications

For scoring matches and mismatches, we define the similarity between two protein pairs as follows:

$$S(uu', vv') = S(u, v)S(u', v') \tag{4.4}$$

$S(uu', vv')$ quantifies the likelihood that the interactions between u and v, and u' and v' are orthologous. Consequently, a match that corresponds to a conserved pair of orthologous interactions is rewarded as follows:

$$\mu(uu', vv') = \bar{\mu}S(uu', vv') \tag{4.5}$$

Here, $\bar{\mu}$ is the match coefficient that is used to tune the relative weight of matches against mismatches and duplications, based on the evolutionary distance between the species that are being compared.

A mismatch may correspond to the functional divergence of either interacting partner after speciation. It might also be due to a false positive or negative in one of the networks that is caused by incompleteness of data or experimental error [26]. However, considering indirect interactions as matches compensates for the second case to a certain extent. According to Wagner [117], after a duplication event, duplicate proteins that retain similar functions in terms of being part of similar processes are likely to be part of the same subnet. Furthermore, since conservation of proteins in a particular *module* is correlated with interconnectedness [72], we expect that interacting partners that are part of a common functional module will at least be linked by short alternate paths. Based on these observations, we penalize mismatches for possible divergence in function as follows:

$$\nu(uu', vv') = -\bar{\nu}S(uu', vv') \tag{4.6}$$

As for match score, mismatch penalty is also normalized by a coefficient $\bar{\nu}$ that determines the relative weight of mismatches with respect to matches and duplications.

While aligning PPI networks, the motivation is to identify conserved patterns of interactions between orthologous proteins. For assessing the likelihood of orthology between proteins, the similarity score defined above relies on sequence homology. However, out-paralogs, which are proteins that are duplicated before the species split hence cannot be considered orthologs, often show sequence similarities as well [120]. Since duplicated proteins rapidly lose their interactions, it is more likely that in-paralogs, *i.e.*, the proteins that are duplicated after a split, will share more interacting partners than out-paralogs do [117]. Therefore, penalizing mismatches implicitly favors *real* orthologs by penalizing the out-paralogs for each interaction that is lost after duplication. Furthermore, we employ sequence similarity as a means for distinguishing in-paralogs from out-paralogs. This is based on the observation that sequence similarity provides a crude approximation for the age of duplication [140]. With the expectation that recently duplicated proteins, which are more likely to be in-paralogs, show more significant sequence similarity than older paralogs, we define duplication score as follows:

$$\delta(u, u') = \bar{\delta}(S(u, u') - \bar{d}) \tag{4.7}$$

Here $\bar{d}$ is the cut-off for being considered in-paralogs. If $S(u, u') > \bar{d}$, suggesting that u and u' are likely to be in-paralogs, the duplication is rewarded by a positive score. If $S(u, u') < \bar{d}$, on the other hand, the proteins are considered out-paralogs, therefore the duplication is penalized.

4.2.2 Alignment Score and the Optimization Problem

The above formulation of match, mismatch, and duplication translates the problem of distinguishing orthologs and in-paralogs from out-paralogs to an optimization problem that accounts for the trade-off between conservation of sequences and interactions. This enables accurate identification of conserved interactions between ortholog

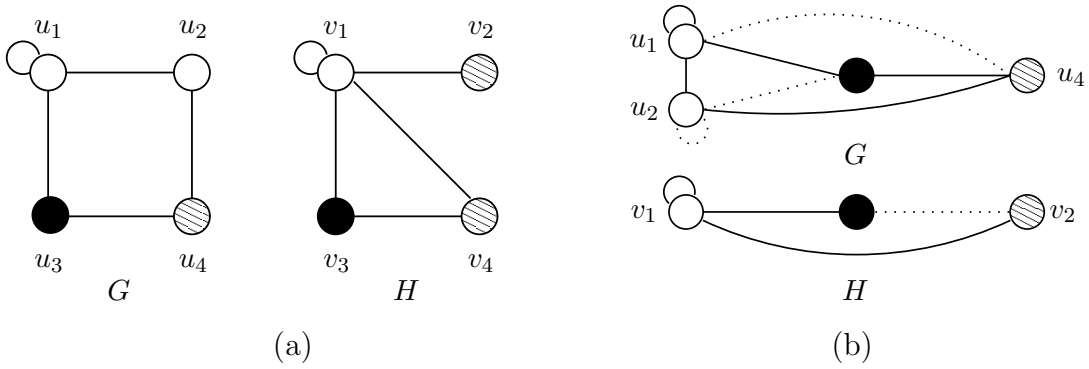


Fig. 4.2. An instance of the pairwise network alignment problem: (a) Two PPI networks, (b) alignment induced by a pair of protein subsets.

protein pairs, while allowing us to define the pairwise local alignment for inter-species comparison of PPI networks as an optimization problem.

Definition 4.2.2 Alignment Score and PPI Network Alignment Problem. *Given PPI networks G and H, the score of alignment $\mathcal{A}(G, H, S, P) = \{\mathcal{M}, \mathcal{N}, \mathcal{D}\}$ is defined as:*

$$\sigma(\mathcal{A}) = \sum_{M \in \mathcal{M}} \mu(M) + \sum_{N \in \mathcal{N}} \nu(N) + \sum_{D \in \mathcal{D}} \delta(D). \tag{4.8}$$

The PPI network alignment problem is one of finding all maximal protein subset pairs P such that $\sigma(\mathcal{A}(G, H, S, P))$ is locally maximal, i.e. the alignment score cannot be improved by adding individual proteins to or removing proteins from P.

We aim to find local alignments with locally maximal score (drawing an analogy to sequence alignment [133], *high-scoring subgraph pair*s).

We illustrate the concepts of match, mismatch, and duplication using a simple example, shown in Figure 4.2. Two sample interaction networks G and H are shown in Figure 4.2(a). The alignment induced by the protein subset pair $\tilde{U} = \{u_1, u_2, u_3, u_4\}$ and $\tilde{V} = \{v_1, v_2, v_3\}$ is shown in Figure 4.2(b), where we set $\bar{\Delta} = 1$. In the figure, the proteins that have non-zero similarity scores (*i.e.*, are potentially orthologous), are colored the same. Note that S does not necessarily induce a disjoint grouping of proteins in practice. Ortholog and paralog proteins are vertically aligned. Existing

interactions are shown by solid lines, missing interactions that have an existing ortholog counterpart are shown by dotted lines. Solid interactions between two aligned proteins in separate species correspond to a match, one solid one dotted interaction between two aligned proteins in separate species correspond to a mismatch. Proteins in the same species that are on the same vertical line correspond to duplications.

The only duplication in this alignment is (u_1, u_2). If this alignment is chosen to be a "good" one, then, based on the existence of this duplication in the alignment, if $S(u_2, v_1) < S(u_1, v_1)$, we can speculate that u_1 and v_1 have evolved from the same gene in the common ancestor, while u_2 is an in-paralog that emerged from duplication of u_1 after split. The match set consists of interaction pairs (u_1u_1, v_1v_1), (u_1u_2, v_1v_1), (u_1u_3, v_1v_3), and (u_2u_4, v_1v_2). Observe that v_1 is mapped to both u_1 and u_2 in the context of different interactions. This is associated with the functional divergence of u_1 and u_2 after duplication. Furthermore, the self-interaction of v_1 in H is mapped to an interaction between paralogous proteins in G.

The mismatch set is composed of $(u_1u_4,\ v_1v_2)$, $(u_2u_2,\ v_1v_1)$, $(u_2u_3,\ v_1v_3)$, and $(u_3u_4,\ v_3v_2)$. The interaction u_3u_4 in G is left unmatched by this alignment, since the only possible pair of proteins in $\tilde{V}$ that are orthologous to these two proteins are v_3 and v_2, which do not interact in H. One conclusion that can be derived from this alignment is the elimination or emergence of this interaction in one of the species after the split. The indirect path between v_3 and v_2 through v_1 may also serve as a basis for the tolerance to the loss of this interaction. Indeed, if we set $\bar{\Delta} = 2$, then this pair of a direct and an indirect interaction would be considered a match. However, if we include v_4 in $\tilde{V}$ as well, then the induced alignment is able to match u_3u_4 and v_3v_4. This strengthens the likelihood that this interaction existed in the common ancestor. However, v_4 comes with another duplication since it is paralogous to v_2. Hence, if $S(v_2, v_4) > \bar{d}$, the alignment that includes v_4 will be favored over the present one. However, if $S(v_2, v_4) < \bar{d}$, then v_4 must compensate for the duplication penalty with the strength of its matching interactions in order to be included in the alignment.

4.2.3 Estimation of Similarity Scores

The similarity score $S(u, v)$ quantifies the likelihood that proteins u and v are orthologous. We approximate this likelihood using the BLAST [1] E-value taking existing ortholog databases as point of reference. Let $\mathbf{O}$ be the set of all orthologous protein pairs derived from COG, or any other ortholog database. For proteins u and v with BLAST E-value $E(u, v) < \tilde{E}$, we approximate the probability of u and v being orthologous by

$$S(u, v) = P(E(u, v) < \tilde{E} | O_{uv}) = \frac{|\{u'v' \in \mathbf{O} : E(u', v') < \tilde{E}\}|}{|\mathbf{O}|} \tag{4.9}$$

where O_{uv} represents the event that u and v are orthologous. If we assume that the probability of a protein pair being orthologous ($P(O_{uv})$) is a monotonically decreasing function of the E-value, then this quantity provides a measure of the likelihood that two proteins with E-value $\tilde{E}$ are orthologous.

4.3 Alignment Graph and Maximum Weight Induced Subgraph Problem

It is possible to represent information regarding matches and mismatches between two PPI networks using a single alignment graph. This graph is a modified version of the graph Cartesian product that takes orthology into account. Assigning appropriate weights to the edges of the alignment graph, the local alignment problem defined in the previous section can be reduced to an optimization problem on this alignment graph. We define alignment graph as follows:

Definition 4.3.1 Alignment Graph.
For a pair of PPI networks $G(U, E)$, $H(V, F)$, and protein similarity function S, the corresponding weighted alignment graph $\mathbf{G}(\mathbf{V}, \mathbf{E})$ is computed as follows:

$$\mathbf{V} = \{\mathbf{v} = \{u, v\} : u \in U, v \in V \text{ and } S(u, v) > 0\}. \tag{4.10}$$

In other words, we have a node in the alignment graph for each pair of ortholog proteins. Each edge $\mathbf{vv'} \in \mathbf{E}$*, where* $\mathbf{v} = \{u, v\}$ *and* $\mathbf{v'} = \{u', v'\}$*, is assigned weight*

$$w(\mathbf{vv'}) = \mu(uu', vv') + \nu(uu', vv') + \delta(u, u') + \delta(v, v'). \tag{4.11}$$

Here, $\mu(uu', vv') = 0$ *if* $(uu', vv') \notin \mathcal{M}$*, and similarly for mismatches and duplications.*

Note that the *alignment graph* is conceptually equivalent to the *global alignment graph* of [134] and the *orthology graph* of [81], with slight differences in formulation. In all models, the nodes of the alignment/orthology graph is constructed from any pair of potentially orthologous proteins in the two networks. On the other hand, in the above-defined alignment graph, all evolutionary information is encoded into edge weights through the concepts of matches, mismatches, and duplications.

Consider the PPI networks in Figure 4.2(a). To construct the corresponding alignment graph, we first compute the product of these two PPI networks to obtain five nodes that correspond to five ortholog protein pairs. We then insert an edge between two nodes of this graph if the corresponding proteins interact in both networks (*match edge*), interact in only one of the networks (*mismatch edge*), or at least one of them is paralogous (*duplication edge*), resulting in the alignment graph shown in Figure 4.3(a). Note that match scores, mismatch and duplication penalties are functions of incident nodes, which is not explicitly shown in the figure for simplicity. Observe that the edge between $\{u_1, v_1\}$ and $\{u_2, v_1\}$ acts a match and duplication edge at the same time, allowing analysis of the conservation of self-interactions of duplicated proteins. This construction of the alignment graph allows us to formulate the alignment problem as a graph optimization problem defined below.

Definition 4.3.2 Maximum Weight Induced Subgraph Problem (MaWISh). *Given graph* $\mathbf{G}(\mathbf{V}, \mathbf{E})$ *and a constant* ϵ*, find a subset of nodes,* $\tilde{\mathbf{V}} \in \mathbf{V}$ *such that the sum of the weights of the edges in the subgraph induced by* $\tilde{\mathbf{V}}$ *is at least* ϵ*, i.e.,* $W(\tilde{\mathbf{V}}) = \sum_{\mathbf{v},\mathbf{v'} \in \tilde{\mathbf{V}}} w(\mathbf{vv'}) \geq \epsilon$.

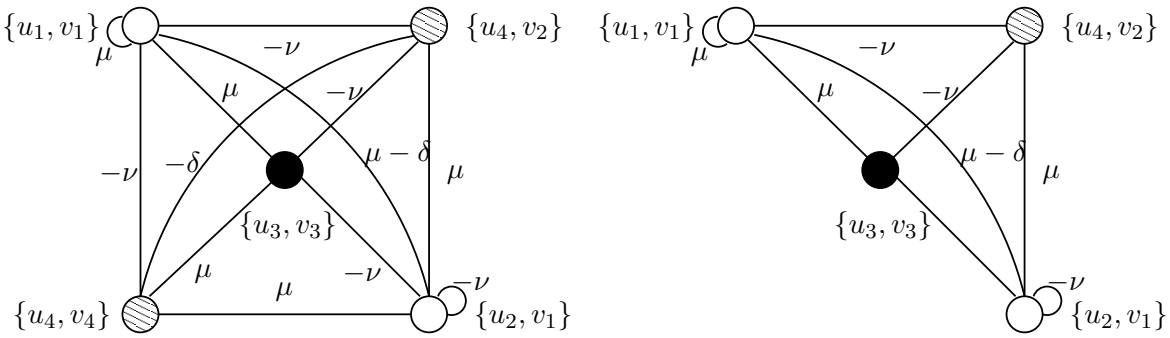


Fig. 4.3. Illustration of alignment graphs: (a) Alignment graph that represents the instance in Figure 4.2(a), (b) a subgraph of this alignment graph, which corresponds to the alignment in Figure 4.2(b).

Not surprisingly, this problem is equivalent to the decision version of the local alignment problem defined in the previous section, as formally stated in the following theorem:

Theorem 4.3.1 *Given PPI networks G, H, and a protein similarity function S, let $\mathbf{G}(\mathbf{V}, \mathbf{E}, w)$ be the corresponding alignment graph. If $\tilde{\mathbf{V}}$ is a solution to the maximum weight induced subgraph problem on $\mathbf{G}(\mathbf{V}, \mathbf{E}, w)$, then $P = \{\tilde{U}, \tilde{V}\}$ induces an alignment $\mathcal{A}(G, H, S, P)$ with $\sigma(\mathcal{A}) = W(\tilde{\mathbf{V}})$, where $\tilde{U} = \{u \in U : \exists v \in V \text{ s.t. } \{u, v\} \in \tilde{\mathbf{V}}\}$ and $\tilde{V} = \{v \in V : \exists u \in U \text{ s.t. } \{u, v\} \in \tilde{\mathbf{V}}\}$.*

Proof. Follows directly from the construction of alignment graph. □

The induced subgraph that corresponds to the local alignment in Figure 4.2(b), $\tilde{\mathbf{V}} = \{\{u_1, v_1\}, \{u_2, v_1\}, \{u_3, v_3\}, \{u_4, v_2\}\}$, is shown in Figure 4.3(b). Note that the weights assigned to these edges, which are shown in the figure, are not constant, but are functions of their incident nodes.

It can be shown that MaWISh is NP-complete by reduction from maximum-clique, by assigning unit weight to edges and $-\infty$ to non-edges. This problem is closely related to the maximum edge subgraph [145] and maximum dispersion problems [146], which are also NP-complete. However, the positive weight restriction on

these problems limits the application of existing algorithms to the maximum weight induced subgraph problem. Nevertheless, the local PPI network alignment problem aims to find all locally maximal alignments, consequently, locally optimal solutions of MaWISh are sufficient. Observing the similarity between min-cut graph partitioning and MaWISh, we develop fast heuristics based on common graph partitioning algorithms to identify locally maximal heavy subgraphs in the alignment graph.

4.4 Algorithms for Alignment of PPI Networks

In terms of protein-protein interactions, functional modules are likely to be densely connected while being separable from other modules, *i.e.*, a protein in a particular module interacts with most proteins in the same module either directly or through a common module hub, while it is only loosely connected to the rest of the network [92]. Since analysis of conserved motifs reveals that proteins in highly connected motifs are more likely to be conserved, suggesting that such dense motifs are parts of functional modules [72], high-scoring local alignments are likely to correspond to functional modules. Therefore, in the alignment graph, we can expect that proteins that belong to a conserved module will induce heavy subgraphs, while being loosely connected to other parts of the graph. This observation motivates the process of greedily growing a subgraph seeded at heavy nodes. This approach is shown to perform well in discovering conserved [81] or dense [73] subnets in PPI networks.

For min-cut graph partitioning, the most commonly applied heuristics are based on starting with a seed partition and repeatedly moving or swapping nodes with maximum gain on the objective function [147]. The key point here is that the move is performed even if it is associated with a negative gain in order to climb over poor local optima. Observe that minimizing the total weight of the cut edges (min-cut) in graph partitioning is equivalent maximizing the total weight of internal edges. This is very similar to the objective function of MaWISh. The difference is that the total weight of only one part is considered in MaWISh, and node balance is not an issue.

Therefore, we apply this iterative improvement based heuristic to MaWISh in order to find locally maximal heavy subgraphs. The initial heavy subgraph is constructed by selecting the node with maximum number of matched interactions (*i.e.*, a *conserved hub*) and adding all nodes that share a match edge with this node to the subgraph.

A sketch of this iterative improvement based algorithm for finding a single conserved subgraph on the alignment graph is shown in Figure 4.4. The procedures Insert, ExtractMax, and Update used in the figure are typical priority queue routines. Each pass (*i.e.*, the loop between lines 3-14) of this algorithm works in linear time. In practice, we also limit the number of contiguous moves with negative gain. This allows us to tune the locality of identified patterns.

To find all non-redundant heavy subgraphs, we start with the entire alignment graph and find a maximally heavy subgraph. We then record the alignment that corresponds to this subgraph and mark its nodes. We repeat this process by considering only unmarked nodes. Once a new heavy subgraph is identified, we add the previously marked nodes that are positively connected to this subgraph one by one, unless the resulting subgraph becomes redundant. A subgraph is said to be redundant if there exists a subgraph which contains $r\%$ of its nodes, where r is a user-defined threshold that determines the extent of allowed overlap. This method allows identification of overlapping alignments while avoiding redundancy. Finally, we rank all subgraphs based on their score and report the corresponding alignments.

4.5 Extensions to the Model

The proposed model can be extended to account for data quality as well as algorithm parameters.

4.5.1 Accounting for Experimental Error

PPI networks obtained from high-throughput screening are prone to errors in terms of both false negatives and positives [26]. While the proposed framework can

procedure HEAVIESTSUBGRAPH($\mathbf{G}$)

▷ **Input** $\mathbf{G}(\mathbf{V}, \mathbf{E}, w)$: Alignment graph

▷ **Output** $\tilde{\mathbf{V}}$: Subset of nodes that induces a maximally heavy subgraph in $\mathbf{G}$

```
1  ṽ ← argmax_{v∈V} |{v' ∈ V : (v, v') is a match edge}|
2  Ṽ ← {ṽ} ∪ {v ∈ V : (ṽ, v) is a match edge}
3  repeat
4      for v ∈ Ṽ do key(v) ← −Σ_{v'∈Ṽ} w(v, v'), INSERT(Q,v)
5      for v ∈ V \ Ṽ do key(v) ← Σ_{v'∈Ṽ} w(v, v'), INSERT(Q,v)
6      W_max ← W(Ṽ)
7      while Q ≠ ∅
8          v ← EXTRACTMAX(Q)
9          if v ∈ Ṽ then Ṽ ← Ṽ \ {v} else Ṽ ← Ṽ ∪ {v}
10         if W(Ṽ) > W_max then W_max ← W(Ṽ), bestmove ← v
11         for all v' such that vv' ∈ E do UPDATE(key(v'))
12     endwhile
13     roll back all moves after bestmove
14 until bestmove = NULL
15 return Ṽ
```

Fig. 4.4. Heuristic algorithm for finding maximal weight induced subgraphs.

be used to detect experimental errors through cross-species comparison to a certain extent, experimental noise can also degrade the performance of the alignment algorithm. In other words, mismatches should be penalized for lost interactions during evolution, not for experimental false negatives. To account for such errors while

analyzing interaction networks, several methods are developed to quantify the likelihood of an interaction or complex co-membership between proteins [35–37]. Given the prior probability distribution for protein interactions and a set of observed interactions, these methods compute the posterior probability of interactions based on Bayesian models. Hence, PPI networks can be modeled using weighted graphs to account for experimental error more accurately.

While the network alignment framework introduced in Section 4.2 assumes that interactions are represented by unweighted edges, it can be easily generalized to a weighted graph model as follows. Assuming that weight ϖ_{uv} represents the posterior probability of interaction between u and v, we can define match score and mismatch penalty in terms of their expected values derived from these posterior probabilities. Therefore, for any $u, u' \in U$ and $v, v' \in V$, we have

$$\mu(uu', vv') = \bar{\mu} S(uu', vv') \varpi_{uu'} \varpi_{vv'} \tag{4.12}$$

$$\nu(uu', vv') = \bar{\nu} S(uu', vv') (\varpi_{uu'}(1 - \varpi_{vv'}) + (1 - \varpi_{uu'})\varpi_{vv'}). \tag{4.13}$$

Note that match and mismatch sets are not necessarily disjoint here in contrast to the unweighted graph model, which is a special case of this model.

4.5.2 Alternate Model Components and Parameters

Contracting Paralogs. An alternate approach to handling duplications is contracting the proteins in the same species that are likely to be in-paralogs. This approach fits into the alignment graph model since in-paralogs are expected to be consistently orthologous to the same set of proteins in the other organism. It also reduces the computational complexity since the number of nodes is decreased by node contraction and the edges that correspond to duplications are eliminated. As also shown in the previous chapter, contraction of nodes is effective for multiple alignment of metabolic pathways using graph mining [16]. However, clustering proteins in the same organism to identify in-paralogs requires preprocessing to solve a difficult problem. Clustering

algorithms that are specifically designed for this purpose, such as INPARANOID [120] serve as reliable tools. However, the resulting graphs may produce conservative alignments since the search space is narrowed down by the clustering of proteins [18]. In contrast, accounting for duplications using duplication edges provides more flexibility and uses conservation of interactions as additional information to distinguish in-paralogs from out-paralogs, as discussed above.

Shortest-path mismatch model. In the above discussion, while we consider proteins that are linked by at most $\bar{\Delta}$ interactions as interacting, we do not take into account the distance while penalizing mismatches. We can extend this to a shortest-path mismatch model, defined as follows:

$$\nu(uu', vv') = \bar{\nu} S(uu', vv')(\max\{\Delta_G(u, u'), \Delta_H(v, v')\} - \bar{\Delta}), \tag{4.14}$$

While this model may improve the alignment algorithm, it is computationally expensive since it requires solution of the all pairs shortest path problem on both PPI networks.

Linear duplication model. The alignment graph model forces each duplicate pair in an alignment to be scored. For example, if an alignment contains n paralogous proteins in one species, $\binom{n}{2}$ duplications are scored to account for each duplicate pair. However, in the evolutionary process, each paralogous protein is the result of a single duplication, *i.e.*, n paralogous proteins are created in only $n-1$ duplications. Therefore, we refer to the current model as *quadratic duplication model*, since the number of scored duplications is a quadratic function of number of duplicates. While this might be desirable as being more restrictive on duplications, to be more consistent with the underlying biological processes, it can be replaced by a *linear duplication model*. In this model, each duplicate protein is penalized only once, based on its similarity with the paralog that is most similar to itself. This model can be incorporated into the alignment graph model of Section 4.4 with a simple modification of the algorithm that dynamically reassigns weights to edges that correspond to duplications.

Table 4.1
Description of three eukaryotic PPI networks obtained from DIP and BIND databases.

Organism	# Proteins	# Interactions
S. cerevisiae	5157	18192
C. elegans	3345	5988
D. melanogaster	8577	28829

4.6 Experimental Results

4.6.1 Data and Implementation

We implement the proposed algorithms in the C programming language and test on PPI networks that belong to three commonly studied eukaryotic organisms. The source code of the software is available at `http://www.cs.purdue.edu/homes/koyuturk/mawish/` along with detailed alignment results. The interaction data is downloaded from BIND [10] and DIP [12] molecular interaction databases. The statistics for the PPI networks of *S. cerevisiae* (yeast), *C. elegans* (nematode), and *D. melanogaster* (fruit fly) are shown in Table 4.1.

We align all pairs of these three organisms using a fixed set of parameters to be able to compare the results with each other. We set these parameters conservatively in order to obtain a compact set of illustrative results. For any pair of PPI networks, we set the *E*-value threshold adaptively based on the estimated similarity scores so that the minimum similarity score for any pair of potential orthologs is 0.6. In other words, two proteins that belong to two different species are considered potentially orthologous only if they have a BLAST *E*-value less than 60% of ortholog pairs in COG. On the other hand, we set $\bar{d} = 0.9$, *i.e.*, two proteins in the same organism are considered potential in-paralogs only if they have BLAST *E*-value less than 90% of protein pairs in this organism that are in the same COG. For potential out-paralogs,

Table 4.2
Alignment statistics for the pairwise alignment of three eukaryotic organisms, *S. cerevisiae* (SC), *C. Elegans* (CE), and *D. melanogaster* (DM).

Organism pair	# Node	# Matched node		# Match		# Mismatch	# Dup.	
		$\bar{\Delta}=1$	$\bar{\Delta}=2$	$\bar{\Delta}=1$	$\bar{\Delta}=2$	$\bar{\Delta}=1$	Org.1	Org.2
SC *vs.* CE	2746	312	1230	412	3007	40262	6107	6886
SC *vs.* DM	15884	1730	8622	2061	42781	1054241	6107	32670
CE *vs.* DM	11805	491	3391	455	6626	205593	6886	32670

we consider protein pairs that have a BLAST *E*-value less than 0.1 but greater than 10% of the ortholog pairs in COG. By setting these cut-off values on similarity score, we only consider the homologous protein pairs that have the highest positive or negative contribution on the alignment score. This eliminates noise to a certain extent while improving the computational efficiency. However, for more detailed analysis and discovery of loosely visible patterns, it may be necessary to relax and set these parameters based on the evolutionary distance between the two organisms being compared.

4.6.2 Results and Discussion

We perform pairwise alignment of the three PPI networks by tuning the alignment parameters to $\bar{\mu} = 1.0$, $\bar{\nu} = 1.0$, and $\bar{\delta} = 0.1$. Detailed statistics on alignment of the three pairs of eukaryotic PPI networks are shown in Table 4.2. In the table, the number of nodes in alignment graphs (# of orthologous pairs), number of nodes with at least one matched edge, number of matches, number of mismatches and number of duplications for both organisms are shown for each alignment. Number of mismatches for $\bar{\Delta} = 2$ can be derived from other statistics. The number of matches and the number matched nodes are shown for two values of $\bar{\Delta}$, where only

direct interactions $\bar{\Delta} = 1$ and indirect interactions through a single protein $\bar{\Delta} = 2$ are considered as matches. In practice, we eliminate all nodes that do not have any matching interactions from the alignment graph. As evident in the table, this improves the computational performance of the algorithm significantly.

Note that the parameters are set in an ad-hoc manner for all results reported in this section, by manually adjusting the balance between normalizing parameters through repeated runs of the algorithm. A more reliable method for adjusting these parameters is to employ a learning heuristic that tunes these parameters in such a way that the conservation/divergence of known functional modules is captured by these parameters, as in the case of sequence alignment. However, this requires knowledge of biologically validated functional modules for both species being compared. We anticipate that as such data becomes available, an accepted set of parameters can easily be derived.

Alignment of *S. cerevisiae* PPI network with *D. melanogaster* PPI network results in identification of 412 conserved subnets. Ten of the conserved subnets with highest alignment scores are shown in Table 4.3. In this table, the rank of the identified subnet among all conserved subnets, the number of alignment nodes and corresponding number of proteins in each organism, total number of matches, mismatches, and duplications in each organism are shown in a row. In the subsequent row, we report the most dominant biological process identified according to the GO annotations of the proteins that are in the conserved network, along with the percentage of proteins that are associated with that biological process, for each organism. Similarly, sample high-scoring conserved subnets identified by the alignment of *S. cerevisiae vs. C. elegans* and C.Elegans *vs. D. melanogaster* PPI networks are shown in Tables 4.4 and 4.5, respectively. In total, 83 conserved subnets are identified on *S. cerevisiae* and *C. elegans*, and 146 are identified on *C. elegans* and *D. melanogaster*.

While most of the conserved subnets are dominated by one particular processes and the dominant processes are generally consistent across species, there also exist different processes in different organisms that are mapped to each other by the dis-

Table 4.3
Sample conserved subnets identified by the alignment of *S. cerevisiae* and *D. melanogaster* PPI networks.

Rank	Score	# Proteins	# Matches	# Mismatches	# Duplications
1	15.97	18 (16, 5)	28	6	(4, 0)
	protein amino acid phosphorylation (69%) /JAK-STAT cascade (40%)				
2	13.93	13 (8, 7)	25	7	(3, 1)
	endocytosis (50%) / calcium-mediated signaling (50%)				
5	8.22	9 (5, 3)	19	11	(1, 0)
	invasive growth (sensu Sacc.) (100%) / O_2 & reactive O_2 species met. (33%)				
6	8.05	8 (5, 3)	12	2	(0, 1)
	ubiquitin-dependent protein catabolism (100%) / mitosis (67%)				
8	6.83	6 (4, 4)	12	6	(0, 1)
	protein amino acid phosphorylation (50%, 50%)				
10	6.75	10 (7, 3)	24	12	(0, 1)
	ubiquitin-dependent protein catabolism (100%)				
14	5.69	11 (11, 2)	10	1	(0, 0)
	regulation of progression through cell cycle (9%, 50%)				
21	4.36	9 (5, 4)	18	13	(0, 5)
	cytokinesis (100%, 50%)				
22	4.22	7 (6, 6)	9	5	(1, 1)
	protein folding (67%, 17%)				
30	3.76	6 (3, 5)	5	1	(0, 6)
	DNA replication initiation (100%, 80%)				

covered alignments. This illustrates that the comparative analysis of PPI networks is effective in not only identifying particular functional modules, pathways, and complexes, but also in discovering relationships between different processes in separate organisms and crosstalk between known functional modules and pathways.

Table 4.4
Sample conserved subnets identified by the alignment of *S. cerevisiae* and *C. elegans* PPI networks.

Rank	Score	# Proteins	# Matches	# Mismatches	# Duplications
1	36.14	13 (5, 3)	65	24	(0, 3)
	ubiquitin-dependent protein catabolism (100%) / reproduction (100%)				
2	8.47	20 (11, 5)	19	4	(1, 1)
	protein amino acid phosphorylation (82%, 40%)				
3	6.28	8 (6, 3)	21	12	(0, 0)
	ubiquitin-dependent protein catabolism (100%, 100%)				
7	3.23	7 (7, 6)	7	2	(0, 0)
	glyoxylate cycle (14%, 17%)				
8	3.23	4 (3, 3)	4	1	(1, 1)
	mismatch repair (67%, 67%)				

A selection of interesting conserved subnets is shown in Figure 4.5. In the figure, orthologous and paralogous proteins are either vertically aligned or connected by dotted lines. Existing interactions are shown by solid lines, missing interactions that have an orthologous counterpart are shown by dashed lines. The rank of each alignment in the set of alignments discovered for the respective pair of organisms is indicated in its label. The alignments in the figure illustrate that the alignment algorithm takes into account the conservation of interactions in addition to sequence similarity while mapping orthologous proteins to each other. In all of the alignments shown in the figure, the interactions of proteins that belong to the same orthologous group are highly conserved, suggesting relatively recent duplications.

Detailed examination of the conserved subnets in *S. cerevisiae* and *D. melanogaster* shows that many of them do correspond to functional modules. There are multiple instances of 20S proteosome. All seven of the alpha subunits in the 20S proteosome,

Table 4.5
Sample conserved subnets identified by the alignment of *C. elegans* and *D. melanogaster* PPI networks.

Rank	Score	# Proteins	# Matches	# Mismatches	# Duplications
1	26.75	17 (4, 9)	52	4	(1, 4)
	thermosensory behav. (25%) / reg. of transc. RNA polymerase II prom. (44%)				
2	4.65	9 (5, 3)	8	0	(2, 1)
	translational initiation (60%, 67%)				
4	4.37	11 (3, 6)	10	1	(1, 4)
	determination of adult life span (33%, 67%)				
5	4.29	5 (4, 4)	6	0	(1, 1)
	regulation of transcription, DNA-dependent (50%, 25%)				
6	4.00	6 (4, 6)	8	2	(0, 2)
	signal transduction (50%, 17%)				

a subcomplex of the 26S proteosome involved in protein degradation, are present in the alignment #10 [148]. In addition, there is a subnet for the proteosome regulatory particle (6) [149] as well as one for calcium induced pathways (2).

The method also detected a number of components involved in calcium-dependent stress-activated signaling pathways (Cmd1, Cna1, Cna2 and Cnb1) as well as those associated with budgrowth of yeast (Cmd1, Myo2 and Myo4) in alignment #2 [150]. Many of the subnets found for yeast are overlapping, possibly reflecting the fact that drosophila uses a functional module in various contexts.

In some cases, the self-interaction of a single protein in one organism is aligned with a clique of interactions between its orthologs that are part of a particular module. For example, in one alignment, five proteosome regulatory particle proteins (Rpt1, Rpt3, Rpt4, Rpt5, Rpt6) are mapped to one protein (Rpt4) in drosophila, while in other alignments the same group of proteins are mapped to a different set of proteins

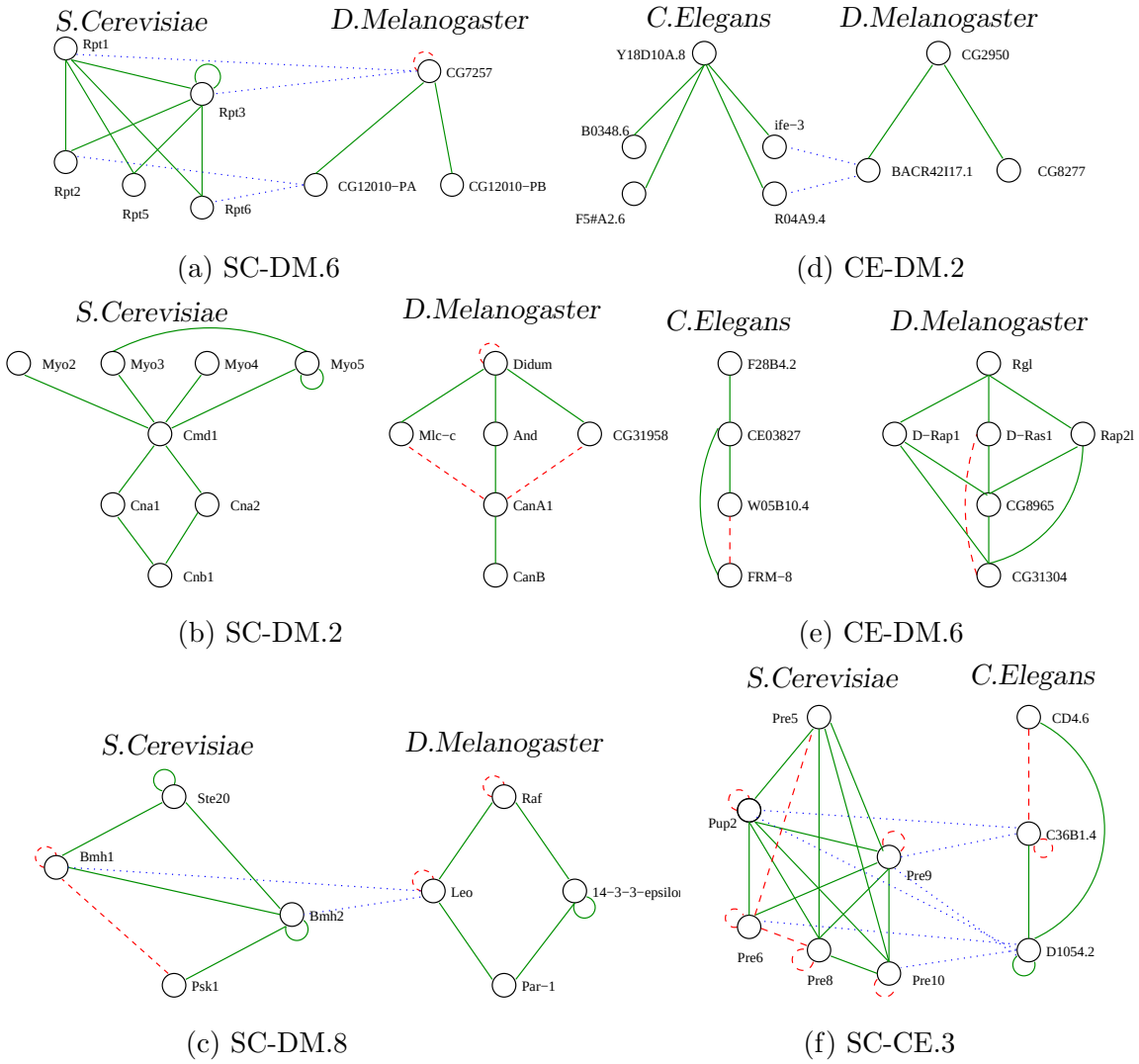


Fig. 4.5. Sample conserved subnets identified by MaWISh.

in the drosophila network. This may be due to missing interactions in one of the networks because of incompleteness or irregularities in the interaction data, including coding of observed interactions into the databases (e.g., spoke *vs.* matrix model [30]). By adjusting the mismatch and duplication coefficients, however, it is possible to make positive duplications dominate the negative mismatches caused by these missing interactions. This may be considered as a desirable feature of our algorithm, in the sense that it allows flexibility for trading off conservation of interactions with

conservation of proteins. However, it might also be considered a drawback, since setting the parameters in this manner causes over-representation of somewhat distant proteins in one conserved subnet on another side of the network.

Based on these results, we establish pairwise alignment of PPI networks as a tool for not only identifying conserved modules, but also assessing functional differences and similarities of homologous proteins based on shared and missing interactions. Furthermore, alignment results provide a means for discovery of new functional modules in relatively less studied organisms through mapping of functions at a modular level rather than at the level of single protein homologies.

5. STATISTICAL SIGNIFICANCE OF CONNECTIVITY AND CONSERVATION IN BIOLOGICAL NETWORKS

There are two critical aspects of identifying meaningful structures in data – the algorithm for the identification and a method for scoring an identified pattern. In this context, the score of a pattern corresponds to its significance. A score is generally computed with respect to a reference model – i.e., given a pattern and a reference model, how likely is it to observe the pattern in the reference model. The less likely such an occurrence is in the reference model, the more interesting it is, since it represents a significant deviation from the reference (nominal) behavior. One such score, in the context of sequences is the E-value returned by BLAST matches [151]. This score broadly corresponds to the likelihood that a match between two sequences is generated by a random process. The lower this value, the more meaningful the match. It is very common in a variety of applications to use a threshold on E-values to identify homologies across sequences. It is reasonable to credit E-value as one of the key ingredients of the success of sequence matching algorithms and software.

While significant progress has been made towards developing algorithms on graphs for identifying patterns (motifs, dense components), conservation, alignment, and related problems, analytical methods for quantifying the significance of such patterns are limited. Existing algorithms for detecting general patterns typically adopt simple ad-hoc measures (such as frequency or relative density) [16, 63], compute z-scores for *the* observed pattern based on simplifying assumptions [18, 81, 82], or rely on Monte-Carlo simulations [81] to assess the significance of identified patterns. Itzkovitz et al. [87] analyze the expected number of occurrences of specific *topological motifs* in a variety of random networks. This study represents, to the best of our knowledge, the first effort at analytically quantifying the statistical significance of the *existence* of *a* pattern with observed property, with respect to a reference model. Specifically,

it presents a framework for analyzing the occurrence of dense patterns in randomly generated graph-structured data (based on the underlying model) with a view to assessing the significance of a pattern based on the statistical relationship between subgraph density and size. This result is generalized in a straightforward manner to the problem of assessing statistical significance of matches between two interaction networks.

The selection of an appropriate reference model for data and the method of scoring a pattern or match, are important aspects of quantifying statistical significance. Using a reference model that fits the data very closely makes it more likely that an experimentally observed biologically significant pattern is generated by a random process drawing data from this model. Conversely, a reference model that is sufficiently distinct from observed data is likely to tag most patterns as being significant. Clearly, neither extreme is desirable for good coverage and accuracy. In this paper, we consider two reference models (i) a $G(n, p)$ model of a graph with n nodes, where each pair of nodes has an identical probability, p, of sharing an edge, and (ii) a two level $G(n, p)$ model in which the graph is modeled as two separate $G(n, p)$ graphs with intervening edges. The latter model captures the heavy nodes corresponding to hub proteins, typically observed in PPIs. For these models, we analytically quantify the behavior of the largest dense subgraph and use this to derive a measure of significance. We show that a simple $G(n, p)$ model can be used to assess the significance of dense patterns in graphs with arbitrary degree distribution, with a conservative adjustment of parameters so that the model stochastically dominates a graph generated according to a given distribution. In particular, by choosing p to be maximal, we ensure that the largest dense subgraph in our $G(n, p)$ model stochastically dominates that of a power-law graph. Our two-level $G(n, p)$ model is devised to mirror key properties of the underlying topology of PPI graphs, and consequently yields a more conservative estimate of significance. Finally, we show how existing graph clustering algorithms [152] can be modified to incorporate statistical significance in identification of dense patterns, resulting in an effective module identification algo-

rithm, SiDeS. (SiDeS is available as a standalone application and as a plugin to Cytoscape over the public domain from our lab.) We also generalize our results and methods to the comparative analysis of PPI networks and show how the significance of a match between two networks can be quantified in terms of the significance of the corresponding dense component in a suitable specified product graph.

Our analytical results are supported by extensive experimental results on a large collection of PPI networks derived from BIND [10] and DIP [12]. These results demonstrate that the proposed model and subsequent analysis provide reliable means for evaluating the statistical significance of highly connected and conserved patterns in PPI networks. We also compare the resulting algorithmic technique, SiDeS, with the module identification algorithm, MCODE [63] and show that SiDeS outperforms this algorithm in terms of specificity and sensitivity of identified clusters with respect to GO annotations. The framework proposed here can be extended to include more general networks that capture the degree distribution of PPI networks more accurately, namely power-law [140, 153], geometric [154], or exponential [155] degree distributions.

The rest of this chapter is organized as follows: In the next section, we discuss graph models for PPI networks. We then analyze the behavior of the largest dense subgraph and derive measures for assessing statistical significance of highly connected as well as highly conserved subgraphs in PPI networks. In Section 5.2, we introduce the SiDeS algorithm. Finally, we present and discuss experimental results in Section 5.3.

5.1 Probabilistic Analysis of Dense Subgraphs

Since proteins that are part of a functional module are likely to densely interact with each other, while being somewhat isolated from the rest of the network [92], many commonly used methods focus on discovering dense regions of the network for identification of functional modules or protein complexes [58, 63–65, 74]. Subgraph

density is also central to many algorithms that target identification of conserved modules and complexes [18, 81, 84]. In order to assess the statistical significance of such dense patterns, we analyze the distribution of the largest "dense" subgraph generated by an underlying reference model. Using this distribution, we estimate the probability that an experimentally observed pattern will occur in the network by chance. The reference model must mirror the basic characteristics of experimentally observed networks in order to capture the underlying biological process correctly, while being simple enough to facilitate theoretical and computational analysis.

5.1.1 Modeling PPI Networks

With the increasing availability of high-throughput interaction data, there has been significant effort aimed at modeling PPI networks. The key observation on these networks is that a few central proteins interact with many proteins, while most proteins in the network have few interacting partners [61, 156]. A commonly accepted model that confirms this observation is based on power-law degree distribution [60, 117, 140, 153]. In this model, the number of nodes in the network that have d neighbors is proportional to $d^{-\gamma}$, where γ is a network-specific parameter. It has also been shown that there exist networks that do not possess a power-law degree distribution [157, 158]. In this respect, alternate models that are based on geometric [154] or exponential [155] degree distribution have been also proposed.

While assessing the statistical significance of identified patterns, existing methods that target identification of highly connected or conserved patterns in PPI networks generally rely on the assumption that interactions in the network are independent of each other [18, 81, 134]. Since degree distribution is critical to the generation of interesting patterns, these methods estimate the probability of each interaction based on the degree distribution of the underlying network. These probabilities can be estimated computationally by generating several random graphs with the same degree distribution via repeated edge swaps and counting the occurrence of each edge

in this large collection of random graphs [81]. Alternately, they can be estimated analytically, by relying on a simple random graph model that is based on a given degree distribution [87,159]. In this model, each node $u \in V(G)$ of graph $G = (V, E)$ is associated with expected degree d_u and the probability of existence of an edge between u and v is defined as $P(uv \in E(G)) = d_u d_v / \sum_{u \in V(G)} d(u)$. In order for this function to be a well-defined probability measure for simple graphs, we must have $d_{\max}^2 \leq \sum_{u \in V(G)} d(u)$, where $d_{\max} = \max_{u \in V(G)} d_u$. However, available protein interaction data generally does not conform to this assumption. For example, based on the PPI networks we derive from BIND [10] and DIP [12] databases, yeast *Jsn1* protein has 298 interacting partners, while the total number of interactions in the *S. cerevisiae* PPI network is 18193. Similarly, the *D. Melanogaster* PPI network with 28830 interactions contains a protein (CG12470-PA ORF) with 207 interacting partners. Such problems complicate the analysis of the significance of certain structures for models that are based on arbitrary degree distribution.

While models that assume power-law [140, 153], geometric [154], or exponential [155] degree distributions may capture the topological characteristics of PPI networks accurately, they require more involved analysis and may also require extensive computation for assessment of significance. To the best of our knowledge, the distribution of dense subgraphs, even maximum clique, which forms a special case of this problem, has not been studied for power-law graphs. In this paper, we first build a framework for the simple and well-studied $G(n, p)$ model and attempt to generalize our results to more complicated models that assume heterogeneous degree distribution.

5.1.2 Largest Dense Subgraph

Given graph G, let $F(U) \subseteq E(G)$ be the set of edges in the subgraph induced by node subset $U \subseteq V(G)$. The density of this subgraph is defined as $\delta(U) = |F(U)|/|U|^2$. Note here that we assume directed edges and allow self-loops for sim-

plicity. PPI networks are undirected graphs and they contain self-loops in general, but any undirected network can be easily modeled using a directed graph and this does not impact the asymptotic correctness of the results. We define a ρ-dense subgraph to be one with density *larger* than pre-defined threshold ρ, *i.e.*, U induces a ρ-dense subgraph if $F(U) \geq \rho|U|^2$. For any ρ, we are interested in the number of nodes in the largest ρ-dense subgraph. This is because any ρ-dense subgraph in the observed PPI network with size larger than this value will be "unusual", *i.e.*, statistically significant. Note that maximum clique is a special case of this problem with $\rho = 1$.

We first analyze the behavior of the largest dense subgraph for the $G(n, p)$ model of random graphs. We subsequently generalize these results to the piecewise degree distribution model in which there are two different probabilities of generating edges. In the $G(n, p)$ model, a graph G contains n nodes and each edge occurs independently with probability p.

Let random variable $R_n(\rho)$ be the size of the maximum subset of vertices that induce a ρ-dense subgraph, *i.e.*,

$$R_n(\rho) = \max_{U \subseteq V(G): \delta(U) \geq \rho} |U|. \tag{5.1}$$

The behavior of $R_n(1)$, which corresponds to maximum clique, is well studied for the $G(n, p)$ model and its typical value is shown to be $O(\log_{1/p} n)$ [160]. In the following theorem, we derive a general result for the typical value of $R_n(\rho)$ for any $\rho > p$.

Theorem 5.1.1 *If G is a random graph with n vertices, where every edge exists with probability p and $\rho > p$, then*

$$\lim_{n \to \infty} \frac{R_n(\rho)}{\log n} = \frac{1}{\kappa(p, \rho)} \qquad (a.s.), \tag{5.2}$$

where

$$\kappa(p, \rho) = -H_p(\rho) = \rho \log \frac{\rho}{p} + (1 - \rho) \log \frac{1 - \rho}{1 - p}. \tag{5.3}$$

Here, $H_p(\rho)$ denotes weighted entropy. More precisely,

$$P(R_n(\rho) \geq r_0) \leq O\left(\frac{\log n}{n^{1/\kappa(p,\rho)}}\right), \tag{5.4}$$

where

$$r_0 = \frac{\log n - \log\log n + \log \kappa(p,\rho) - \log e + 1}{\kappa(p,\rho)} \tag{5.5}$$

for large n.

Proof. We first prove the upper-bound. Let $X_{r,\rho}$ denote the number of subgraphs of size r with density at least ρ, *i.e.*, $X_{r,\rho} = |\{U \subseteq V(G) : |U| = r \wedge |F(U)| \geq \rho r^2\}|$. From first moment method, we obtain $P(R_n(\rho) \geq r) \leq P(X_{r,\rho} \geq 1) \leq \mathbf{E}[X_{r,\rho}]$.

Let Y_r denote the number of edges induced by r vertices. Then, $\mathbf{E}[X_r] = \binom{n}{r} P(Y_r \geq \rho r^2)$. Furthermore, since Y_r is a Binomial r.v. $B(r^2, p)$ and $\rho > p$, we have

$$P(Y_r \geq \rho r^2) \leq (r^2 - \rho r^2) P(Y_r = \rho r^2) \leq \binom{r^2}{\rho r^2} (r^2 - \rho r^2) p^{\rho r^2} (1-p)^{r^2 - \rho r^2}. \tag{5.6}$$

Hence, we get $P(R_n(\rho) \geq r) \leq \binom{n}{r}\binom{r^2}{\rho r^2}(r^2 - \rho r^2) p^{\rho r^2}(1-p)^{r^2-\rho r^2}$.

Using Stirling's formula, we find the following asymptotics for $\binom{n}{r}$:

$$\binom{n}{r} \sim \begin{cases} \frac{1}{\sqrt{2\pi r}} \frac{n^r}{r^r} e^{-r} & \text{if } r = o(\sqrt{n}) \\ \frac{1}{\sqrt{2\pi\alpha(1-\alpha)n}} 2^{nH(\alpha)} & \text{if } r = \alpha n \end{cases} \tag{5.7}$$

where $H(\alpha) = -\alpha \log \alpha - (1-\alpha)\log(1-\alpha)$ denotes the binary entropy.

Let $Q = 1/p^\rho (1-p)^{1-\rho}$. Plugging the above asymptotics into (5.1.2), we obtain

$$P(R_n(\rho) \geq r) \leq \frac{r\sqrt{1-\rho}}{2\pi\sqrt{\rho}} \exp_2(-r^2 \log Q + r \log n - r \log r + r^2 H(\rho) - r \log e) \tag{5.8}$$

Defining $\kappa(p,\rho) = \log Q - H(\rho)$, we find $P(R_n(\rho) \geq r_0) \leq \frac{r_0\sqrt{1-\rho}}{2\pi\sqrt{\rho}} \exp_2(f(r_0))$, where $f(r_0) = -r_0(r_0\kappa(p,\rho) - \log n + \log r + \log e)$. Plugging in (5.5) and working out the algebra, we obtain $f(r_0) = -r_0\left(1 - O\left(\frac{\log\log n}{\log n}\right)\right)$. Hence, $P(R_n(\rho) \geq r_0) \leq O\left(2^{-r_0}\right) = O\left(\frac{\log n}{n^{1/\kappa(p,\rho)}}\right)$. This completes the proof for the upper-bound.

For the lower bound, we have

$$P(R_n(\rho) < r) = P(X_{r,\rho} = 0) \leq \frac{\mathbf{E}[X_{r,\rho}^2]}{\mathbf{E}[X_{r,\rho}]^2}. \tag{5.9}$$

from second moment method [161]. Letting $m = \rho r^2$, we obtain $\mathbf{E}[X_{r,\rho}] = \binom{n}{r}\binom{r}{m} p^m$ q^{r^2-m} and

$$\mathbf{E}[X_{r,\rho}^2] = \binom{n}{r} \sum_{l=0}^{r} \binom{r}{l}\binom{n-r}{r-l} \sum_{k\in I_l} \binom{l^2}{k} p^k (1-p)^{l^2-k} \left[\binom{r^2-l^2}{m-k} p^{m-k}(1-p)^{r^2-l^2-(m-k)}\right]^2 \tag{5.10}$$

where $I_l = \{k : \max(0, l^2 + m - r^2) \le k \le \min(l^2, m)\}$. Here, for two node subsets U_r and V_r, l denotes the number of nodes at the intersection of U_r and V_r, *i.e.*, $l = |U_r \cap V_r|$. On the other hand, k denotes the number of edges at the intersection of the subgraphs induced by U_r and V_r, *i.e.*, $k = |F(U_r) \cap F(V_r)|$. Hence,

$$\frac{\mathbf{E}[X_{r,\rho}^2]}{\mathbf{E}[X_{r,\rho}]^2} = \sum_{l=0}^{r} \sum_{k\in I_l} f(r,l,k) \tag{5.11}$$

where

$$f(r,l,k) = \frac{\binom{n-r}{r-l}\binom{r}{l}\binom{l^2}{k}\binom{r^2-l^2}{m-k}^2 p^{-k}(1-p)^{k-l^2}}{\binom{n}{r}\binom{r^2}{m}^2}. \tag{5.12}$$

Therefore,

$$P(R_n(\rho) < r) \le \sum_{l=0}^{r} \sum_{k\in I_l} f(r,l,k) \le r^3 \max_{l,k} f(r,l,k). \tag{5.13}$$

For $r = \frac{(1-\epsilon)\log n}{\kappa(\rho,p)}$, $0 \le l \le r$ and $k \in I_l$, we will show that

$$f\left(\frac{(1-\epsilon)\log n}{\kappa(\rho,p)}, l, k\right) \le n^{\frac{-\epsilon(1-\epsilon)\log n}{\kappa(\rho,p)}} \tag{5.14}$$

to conclude that $P(R_n(\rho) < r) \le \frac{(\log n)^3}{n^{\frac{\epsilon(1-\epsilon)\log n}{\kappa(\rho,p)}}}$. To achieve this, let $\alpha = l^2/r^2$ and $\beta = k/l^2$. Then, assuming $\rho > 1/2$ without loss of generality, the interval corresponding to I_l for $0 \le \alpha \le 1$ becomes

$$J_\alpha = \begin{Bmatrix} 0 \le \beta \le 1 & \text{if } 0 \le \alpha \le 1-\rho \\ \beta : \ \frac{\alpha+\rho-1}{\alpha} \le \beta \le 1 & \text{if } 1-\rho \le \alpha \le \rho \\ \frac{\alpha+\rho-1}{\alpha} \le \beta \le \frac{\rho}{\alpha} & \text{if } \rho \le \alpha \le 1. \end{Bmatrix} \tag{5.15}$$

Inserting $l = \sqrt{\alpha} r$ and $k = \alpha\beta r^2$ in (5.12), we obtain

$$f_{\alpha,\beta}(r) = \frac{\binom{r}{\sqrt{\alpha}r}\binom{n-r}{(1-\sqrt{\alpha})r}\binom{\alpha r^2}{\alpha\beta r^2}\binom{(1-\alpha)r^2}{(\rho-\alpha\beta)r^2}^2 p^{-\alpha\beta r^2}(1-p)^{\alpha(\beta-1)r^2}}{\binom{n}{r}\binom{r^2}{\rho r^2}^2}. \tag{5.16}$$

Plugging Stirling's approximation (5.7) for appropriate regimes, we get

$$\begin{aligned}\log(f_{\alpha,\beta}(r)) \sim \quad & -r(\sqrt{\alpha}\log n) + r^2(\alpha H(\beta) - \alpha(\beta\log p + (1-\beta)\log(1-p)) \\ & +2(1-\alpha)H\left(\tfrac{\rho-\alpha\beta}{1-\alpha}\right) - 2H(\rho)).\end{aligned} \tag{5.17}$$

Hence, for $r = \frac{(1-\epsilon)\log n}{\kappa(p,\rho)}$, we have

$$\log\left(f_{\alpha,\beta}\left(\frac{(1-\epsilon)\log n}{\kappa(p,\rho)}\right)\right) \sim \frac{1-\epsilon}{\kappa(p,\rho)}(\log n)^2\left[-\sqrt{\alpha} + \frac{1-\epsilon}{\kappa(p,\rho)}g(\alpha,\beta)\right] \tag{5.18}$$

where

$$\begin{aligned}g(\alpha,\beta) = \quad & \alpha H(\beta) - \alpha(\beta\log p + (1-\beta)\log(1-p)) \\ & +2(1-\alpha)H\left(\tfrac{\rho-\alpha\beta}{1-\alpha}\right) - 2H(\rho).\end{aligned} \tag{5.19}$$

Working out the algebra, we observe that

$$\max_{0\le\alpha\le 1,\beta\in J_\alpha} g(\alpha,\beta) = g(1,\rho) = \kappa(\rho,p) \tag{5.20}$$

where the maximum corresponds to the boundary point $l = r$ and $k = \rho r^2$. Hence, it immediately follows from (5.18) that $\log(f) \le \frac{-\epsilon(1-\epsilon)}{\kappa(p,\rho)}(\log n)^2$ for $0 \le \alpha \le 1$ and $\beta \in J_\alpha$, which leads to (5.14). □

Observe that, if n is large enough, the probability that a dense subgraph of size r_0 exists in the subgraph is very small. Consequently, r_0 may provide a threshold for deciding whether an observed dense pattern is statistically significant.

For a graph of arbitrary degree distribution, let $d_{\max}$ denote the maximum expected degree as defined in Section 5.1.1. Let $p_{\max} = d_{\max}/n$. It can be easily shown that the largest dense subgraph in the $G(n,p)$ graph with $p = p_{\max}$ stochastically dominates that in the random graph generated according to the given degree distribution (e.g., power-law graphs). Hence, by estimating the edge probability conservatively, we can use the above result to determine whether a dense subgraph identified in a PPI network of arbitrary degree distribution is statistically significant. Furthermore, the above result also provides a means for quantifying the significance of an observed dense subgraph. For a subgraph with size $\hat{r} > r_0$ and density $\hat{\rho}$, let $\epsilon = \frac{\hat{r} - \log n/\kappa(\hat{\rho},p)}{\log n/\kappa(\hat{\rho},p)}$.

Then, it follows from (5.8) that the probability of observing this subgraph in a graph generated according to the reference model is bounded by

$$P(R_{\hat{\rho}} \geq (1+\epsilon)\log n/\kappa(\hat{\rho},p)) \leq \frac{\sqrt{1-\rho}}{2\pi\sqrt{\rho}} \frac{(1+\epsilon)\log n}{n^{\epsilon(1+\epsilon)\log n/\kappa(\hat{\rho},p)}}. \tag{5.21}$$

While these results for the $G(n,p)$ model provide a simple yet effective way of assessing statistical significance of dense subgraphs, we extend our analysis to a more complicated model, which takes into account the degree distribution to capture the topology of the PPI networks more accurately.

5.1.3 Piecewise Degree Distribution Model

In the piecewise degree distribution model, nodes of the graph are divided into two classes, namely high-degree and low-degree nodes. More precisely, we define random graph G with node set $V(G)$ that is composed of two disjoint subsets $V_h \subset V(G)$ and $V_l = V(G) \setminus V_h$, where $n_h = |V_h| \ll |V_l| = n_l$ and $n_h + n_l = n = |V(G)|$. In the reference graph, the probability of an edge is defined based on the classes of its incident nodes as:

$$P(uv \in E(G)) = \begin{cases} p_h & \text{if } u,v \in V_h \\ p_l & \text{if } u,v \in V_l \\ p_b & \text{if } u \in V_h, v \in V_l \text{ or } u \in V_l, v \in V_h \end{cases} \tag{5.22}$$

Here, $p_l < p_b < p_h$. This model captures the key lethality and centrality properties of PPI networks in the sense that a few nodes are highly connected while most nodes in the network have low degree [61, 156]. Observe that, under this model, G can be viewed as a superposition of three random graphs G_l, G_h, and G_b. Here, G_h and G_l are $G(n,p)$ graphs with parameters (n_h, p_h) and (n_l, p_l), respectively. G_b, on the other hand, is a random bipartite graph with node sets V_l, V_h, where each edge occurs with probability p_b. Hence, we have $E(G) = E(G_l) \cup E(G_h) \cup E(G_b)$. This facilitates direct employment of the results in the previous section for analyzing graphs with piecewise degree distribution.

We now show that the high-degree nodes in the piecewise degree distribution model contribute a constant factor to the typical size of the largest dense subgraph as long as n_h is bounded by a constant.

Theorem 5.1.2 *Let G be a random graph with piecewise degree distribution, as defined by (5.22). If $n_h = O(1)$, then*

$$P(R_n(\rho) \geq r_1) \leq O\left(\frac{\log n}{n^{1/\kappa(p_l,\rho)}}\right), \tag{5.23}$$

where

$$r_1 = \frac{\log n - \log\log n + 2n_h \log B + \log \kappa(p_l,\rho) - \log e + 1}{\kappa(p_l,\rho)} \tag{5.24}$$

and $B = \frac{p_b q_l}{p_l} + q_b$, where $q_b = 1 - p_b$ and $q_l = 1 - p_l$.

Proof. Let $X^h_{r,\rho}$, $X^l_{r,\rho}$ be the number of ρ-dense subgraphs induced by only nodes in G_h or G_l, respectively. Let $X^b_{r,\rho}$ be the number of these induced by nodes from both sets. Clearly, $X_{r,\rho} = X^h_{r,\rho} + X^l_{r,\rho} + X^b_{r,\rho}$. The analysis for $G(n,p)$ directly applies for $\mathbf{E}[X^h_{r,\rho}]$ and $\mathbf{E}[X^l_{r,\rho}]$, hence we emphasize on $\mathbf{E}[X^b_{r,\rho}]$. Since $n_h = O(1)$, we have $\mathbf{E}[X^b_{r,\rho}] \leq (1-\rho)r^2 \sum_{k=0}^{n_h} \binom{n_h}{k}\binom{n_l}{r-k} \sum_{l=0}^{2k(r-k)} \binom{2k(r-k)}{l}\binom{(r-k)^2}{\rho r^2 - l} p_b^l q_b^{2k(r-k)-l} p_l^{\rho r^2 - l}$ $q_l^{(r-k)^2 - \rho r^2 + l}$,where $q_b = 1 - p_b$ and $q_l = 1 - p_l$. Then,

$$\mathbf{E}[X^b_{r,\rho}] \leq c(1-\rho)r^2 n_h \binom{n_l}{r} \sum_{l=0}^{2n_h r} \binom{2n_h r}{l}\binom{r^2}{\rho r^2 - l} p_b^l q_b^{2n_h r - l} p_l^{\rho r^2 - l} q_l^{r^2 - \rho r^2 + l}, \tag{5.25}$$

where c is a constant. Since $l = o(\rho r^2)$, we have $\binom{r^2}{\rho r^2 - l} \leq \binom{r^2}{\rho r^2}$ for $0 \leq l \leq 2n_h r$. Therefore,

$$\mathbf{E}[X^b_{r,\rho}] \leq (1-\rho)r^2 \binom{n}{r}\binom{r^2}{\rho r^2} p_l^{\rho r^2} q_l^{r^2 - \rho r^2} \sum_{l=0}^{2n_h r} \binom{2n_h r}{l} \left(\frac{p_b q_l}{p_l}\right)^l q_b^{2n_h r - l}. \tag{5.26}$$

Using $B = \frac{p_b q_l}{p_l} + q_b$ as defined in Theorem 5.1.2, we find $P(R_n(\rho) > r) \leq O(2^{f_1(r)})$, where $f_1(r) = -r(r\kappa(\rho) - \log n + \log r - \log e + 2n_h \log B)$.Hence, $P(R_n(\rho) > r_1) \leq O(2^{f_1(r_1)}) \leq O\left(\frac{\log n}{n^{1/\kappa(p_l,\rho)}}\right)$ for large n. □

Note that the above result is based on asymptotic behavior of r_1, hence the $\log n$ term dominates as $n \to \infty$. However, if n is not large enough, the $2n_h \log B$ term may

cause over-estimation of the critical value of the largest dense subgraph. Therefore, the application of this theorem is limited for smaller n and the choice of n_h is critical.

A heuristic approach for estimating n_h is as follows. Assume that the underlying graph is generated by a power-law degree distribution, where the number of nodes with degree d is given by $nd^{-\gamma}/\zeta(\gamma)$ [162]. Here, $\zeta(.)$ denotes the Riemann zeta-function. If we divide the nodes of this graph into two classes where high-degree nodes are those with degree $d \geq (n/\zeta(\gamma))^{1/\gamma}$ so that the expected number of nodes with degree d is at most one, then $n_h = \sum_{d=(n/\zeta(\gamma))^{1/\gamma}}^{\infty} nd^{-\gamma}/\zeta(\gamma)$ is bounded, provided the above series converges.

5.1.4 Conservation of Dense Subgraphs

Comparative methods that target identification of conserved subnets in PPI networks induce a cross-product, or superposition, of several networks in which each node corresponds to a group of orthologous proteins [17, 18, 81, 82, 134]. Here, we rely on ortholog groups available in the COG database [121] to relate proteins in different PPI networks [17]. Labeling each node in the PPI network with the COG family of the protein it represents, we obtain an intersection of two PPI networks by inserting an edge between two COG families only if proteins that belong to these families interact in both graphs. In the case of the $G(n, p)$ model, the above framework directly applies to the identification of dense subgraphs in this intersection graph, where the probability of observing a conserved interaction is estimated as $p_I = p_1 p_2$. Here p_1 and p_2 denote the probability of observing an edge in the first and second networks, respectively. For the piecewise degree distribution model, on the other hand, we have to assume that the orthologs of high-degree nodes in one graph are high-degree nodes in the other graph as well. If this assumption is removed, it can still be shown that the low-degree nodes dominate the typical behavior of the largest conserved subgraph. Note that the reference model assumes that the orthology relationship between proteins in the two networks is already established and the model estimates

the conditional probability that the interactions between these given ortholog proteins are densely conserved.

5.2 SIDES: An Algorithm for Identification of Significant Dense Subgraphs

We use the above results to modify an existing state-of-the-art graph clustering algorithm, HCS [152], in order to incorporate statistical significance in identification of interesting dense subgraphs. HCS is a recursive algorithm that is based on decomposing the graph into dense subgraphs by recursive application of min-cut partitioning. A min-cut partition of the nodes of a graph $G = (V, E)$ is a disjoint partition of V into V_0 and V_1 such that the cut

$$C(V_0, V_1) = |\{uv \in E : u \in V_0, v \in V_1 \vee u \in V_1, v \in V_0\}| \tag{5.27}$$

is minimized. In the original HCS algorithm, the density of any subgraph found in this recursive decomposition is compared with a pre-defined density threshold. If a subgraph is dense enough, it is reported as a highly-connected cluster of nodes, else it is partitioned again. While this algorithm provides a strong heuristic that is well suited to the identification of densely interacting proteins in PPI networks [56], the selection of density threshold poses an important problem. In other words, it is hard to provide a biologically justifiable answer to the question "How dense must a subnetwork of a PPI network be to be considered biologically interesting?". Our framework provides an answer to this question from a statistical point of view by establishing the relationship between subgraph size and density as a stopping criterion for the algorithm.

For any subgraph encountered during the course of the algorithm, we estimate the critical size of the subgraph to be considered interesting, by plugging in its density in (5.5) or (5.24). If the size of the subgraph is larger than this probabilistic upper-bound, we report the subgraph as being statistically significant. Otherwise, we continue partitioning the graph.

procedure MinCutPhase (S, s)

▷ **Input** S: Subgraph to be partitioned
▷ **Input** $s \in V(S)$: A fixed node of S
▷ $w(uv)$: Number of edges between nodes represented by u and by v
▷ **returns** the last two nodes and the cut between last node and others

1 $\mathcal{V} \leftarrow \{s\}$
2 **while** $|\mathcal{V}| < |V(S)| - 1$ **do**
3 $\quad v \leftarrow \text{argmax}_{v' \in V(S)} \sum_{u' \in \mathcal{V}} w(u'v')$
4 $\quad \mathcal{V} \leftarrow \mathcal{V} \cup \{v\}$
5 $u \leftarrow V(S) \setminus \mathcal{V}$
6 **return** $\{v, u, \sum_{u' \in \mathcal{V}} w(u'u)\}$

Fig. 5.1. A single phase of the min-cut algorithm used by SiDeS.

An important problem relating to the use of min-cut partitioning is that min-cut partitioning tends to single out a node in one part, since no balance constraint is imposed. Hence, recursive application of min-cut on large graph is likely to result in many clusters containing a single node, which indeed is not significant. This problem is particularly important in PPI networks because of their characteristic degree distribution, *i.e.*, most proteins in the network are low-degree nodes, which are likely to be singled out by min-cut partitioning. We resolve this problem by an additional modification to the HCS algorithm and we partition the network to minimize the ratio cut rather than the edge cut. Ratio cut partitioning is a well-studied problem in various contexts. It targets minimization of the edge cut while maintaining balance implicitly, without imposing any strict balance constraints [163].

procedure RatioCutPartition (S)

▷ **Input** S: Subgraph to be partitioned
▷ $w(u)$: Number of nodes represented by u
▷ **returns** partition that locally minimizes ratio-cut

```
1  for u ∈ V(S) do
2      w(u) ← 1
3  W ← |V(S)|
4  R̄ ← E(S) + 1
5  pick arbitrary seed node s ∈ V(S)
6  while |V(S)| > 1 do
7      {v, u, C} ← MinCutPhase(S, s)
8      R = C/ min(w(u), W − w(u))
9      if R < R̄ then R̄ ← R
10     merge u into v, w(v) ← w(v) + w(u)
11 return partition that corresponds to R̄
```

Fig. 5.2. Ratio-cut partitioning algorithm used by SiDeS.

Although being NP-hard, in contrast to the min-cut problem [164], the problem can be solved effectively by heuristic methods and is very well suited for partitioning of PPI networks since no strict balance is required but single-node partitioning needs to be avoided. In our implementation, we define ratio-cut as

$$R(V_0, V_1) = \frac{C(V_0, V_1)}{\min(|V_0|, |V_1|)} \tag{5.28}$$

and adopt a simple min-cut algorithm [165] to heuristically solve this problem. The underlying algorithm considers $|V|$ partitions, which are locally optimal and chooses

procedure RecursiveRatioCut(S, n, p)

▷ **Input** S: Subgraph to be partitioned
▷ **Input** n: Number of nodes in original graph
▷ **Input** p: Probability of existence of an edge
▷ **returns** set of dense subgraphs of S that are significant w.r.t. n and p

```
1  ρ ← |E(S)|/|V(S)|²
2  Estimate r_0 as given by (5.5)
3  if |V(S)| > r_0 then
4     Estimate significance of S as given by (5.21)
5     return {S}
6  else
7     {S_0, S_1} ← RatioCutPartition(S)
8     return RecursiveRatioCut(S_0, n, p) ⋃ RecursiveRatioCut(S_1, n, p)
```

Fig. 5.3. Recursive partitioning algorithm used by SiDeS.

the one that induces minimum edge-cut, which is shown to be the global optimum. In our implementation, we consider the same $|V|$ partitions, but choose the one that minimizes the ratio cut of (5.28) to heuristically favor a more balanced partition.

The resulting *si*gnificant *de*nse *s*ubgraph identification algorithm, SiDeS, is shown in Figures 5.1-5.4. Details of the recursive algorithm and the min-cut algorithm can be found in [152] and [165], respectively. Note that this algorithm only identifies disjoint subgraphs, but can be easily extended to obtain overlapping dense subgraphs by greedily growing each of the resulting subgraphs until significance is lost. The C

procedure SiDeS(G)

▷ **Input** G: Input network

▷ **returns** set of significant dense subgraphs of G

1 $p \leftarrow \max_{u \in V} |\{v \in V(G) : uv \in E(G)\}|/|V(G)|$

2 **return** RecursiveRatioCut(G, $|V(G)|$, p)

Fig. 5.4. SiDeS algorithm for identifying significant dense subgraphs in a network.

source code and a Java implementation as a Cytoscape [166] plug-in for SiDeS are available as open source at `http://www.cs.purdue.edu/homes/koyuturk/sides/`.

5.3 Experimental Results

In this section, we first compare the behavior of dense subgraphs in experimentally available network data with the theoretical results presented in this paper. Then, we present experimental results on the performance of SiDeS, which uses statistical significance as an optimization criterion, and demonstrate the excellent performance of SiDeS in identifying biologically relevant protein clusters as compared to existing algorithms. We do this by quantifying the biological significance of identified clusters in terms of specificity and sensitivity.

5.3.1 Behavior of Largest Dense Subgraph

We experimentally analyze connectivity and conservation in PPI networks of 11 species gathered from BIND [10] and DIP [12] databases. These networks vary signifi-

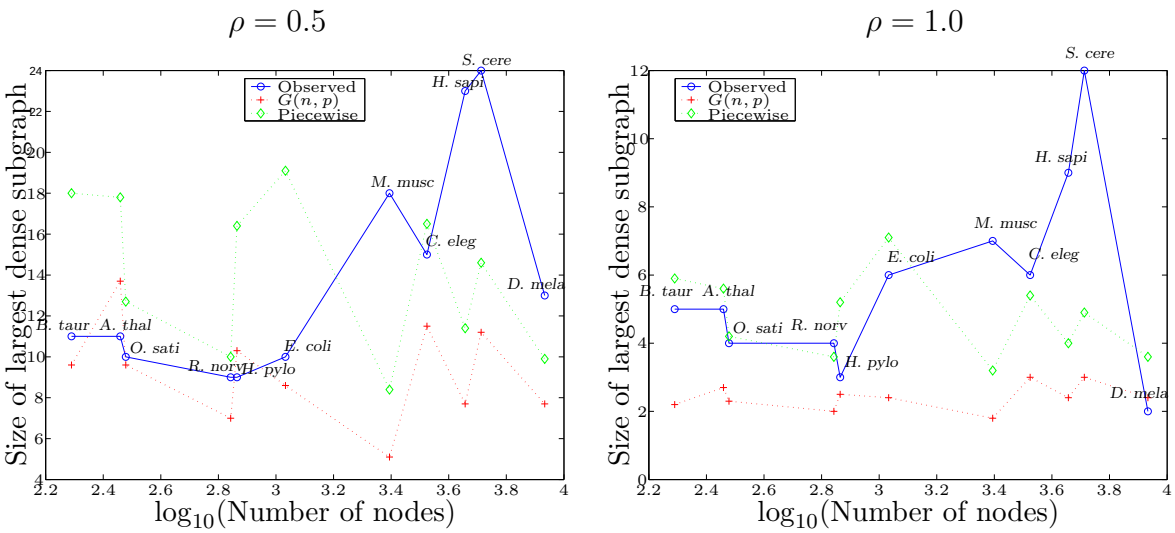


Fig. 5.5. The behavior largest dense subgraph size with respect to number of proteins in the network.

cantly in size and comprehensiveness and cover a broad range of organisms. Relatively large amounts of interaction data is available for *S.cerevisiae* (18192 interactions between 5157 proteins), *D. melanogaster* (28829 among 8577), *H. sapiens* (7393 among 4541), *C. elegans* (5988 among 3345), *E. coli* (1329 among 1079), while the networks for other organisms are restricted to a small portion of their networks.

In Figure 5.5, we examine the behavior of largest subgraph with respect to number of nodes in the PPI network for two different values of density threshold (ρ), 0.5 and 1 (clique). In the figure, each organism corresponds to a sample point, which is marked by its name. The critical values of largest dense subgraph size based on $G(n,p)$ and piecewise degree distribution models are also shown in the figure. Since the sparsity and degree distribution of these networks vary significantly across different organisms, the estimated values of edge probabilities vary accordingly. Hence, the curves for r_0 ($G(n,p)$ model) and r_1 (piecewise degree distribution model) do not show a linear behavior. As seen in the figure, piecewise degree distribution model provides a more conservative assessment of significance. This is primarily because of the constant factor in the critical value of r_1. Observed size of the largest dense subgraph in

smaller networks is not statistically significant, while larger and more comprehensive networks contain subgraphs that are twice as large as the theoretical estimate, with the exception of the *D. melanogaster* PPI network. The lack of dense subnets in the *D. melanogaster* network may be due to differences in experimental techniques (*e.g.*, two hybrid *vs.* AP/MS) and/or the incorporation of identified interactions in the interaction network model (*e.g.*, spoke *vs.* matrix) [30]. In order to avoid problems associated with such variability, it may be necessary to revise the definition of subgraph density or preprocess the PPI networks to standardize the topological representation of protein complexes in the network model.

The behavior of largest dense subgraph size with respect to density threshold is shown in Figure 5.6 for *S. Cerevisiae* and *H. Sapiens* PPI networks and their intersection. It is evident from the figure that the observed size of the largest dense subgraph follows a similar trajectory with the theoretical values estimated by both models. Furthermore, in both networks, the largest dense subgraph turns out to be significant for a wide range of density thresholds. For lower values of ρ, the observed subgraphs are either not significant or are marginally significant. This is a desirable characteristic of significance-based analysis since identification of very large sparse subgraphs should be avoided while searching for dense patterns in PPI networks. Observing that the $G(n, p)$ model becomes more conservative than the piecewise degree distribution model for lower values of ρ, we conclude that this model may facilitate fine-grain analysis of modularity in PPI networks.

5.3.2 Performance of SiDeS

In this section, we demonstrate the performance of SiDeS in identification of significant dense subgraphs on the available yeast PPI network and compare it with an existing complex identification algorithm, MCODE [63]. Both algorithms work on a set of interactions modeled as a simple graph and return a set of protein clusters, each of which induce unusually dense subgraphs in the network. MCODE associates

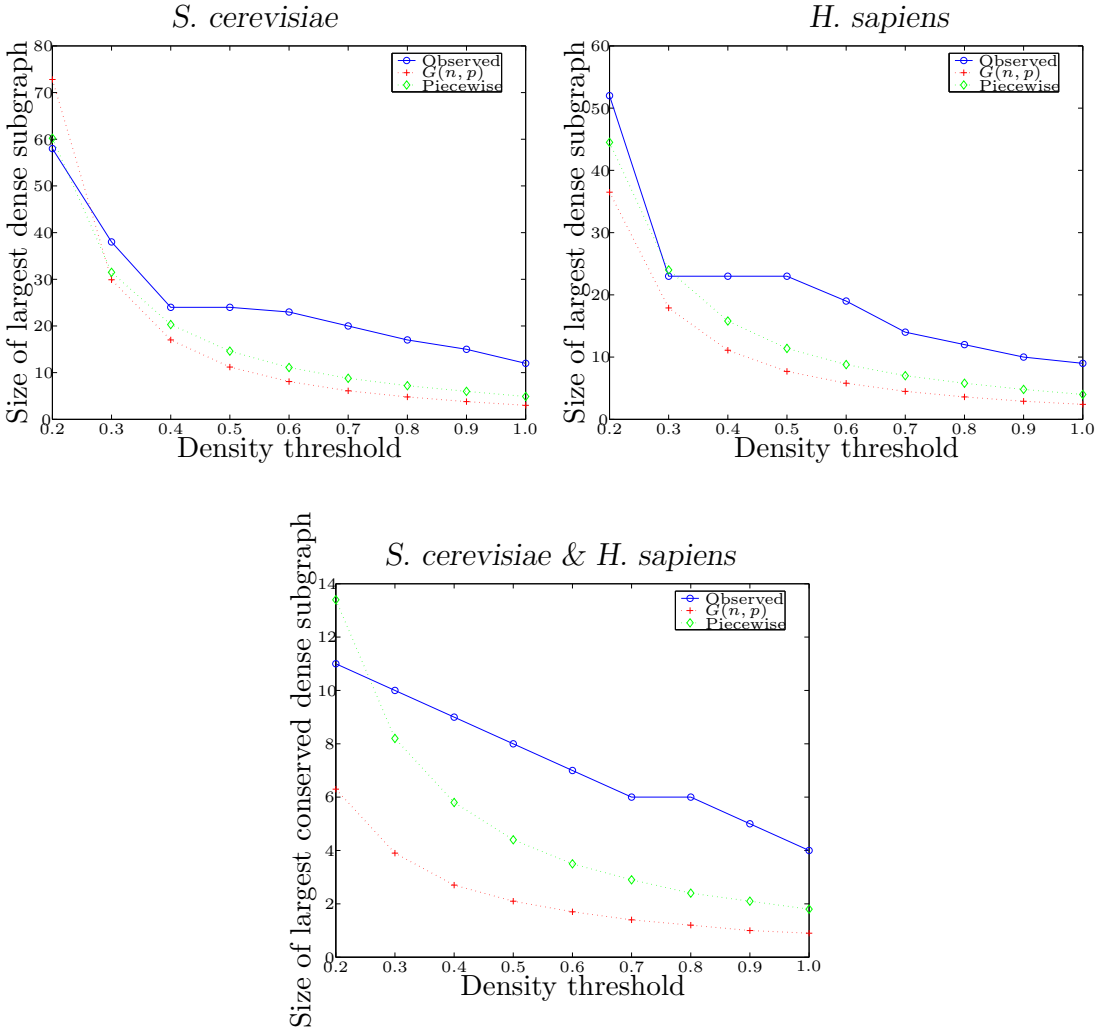
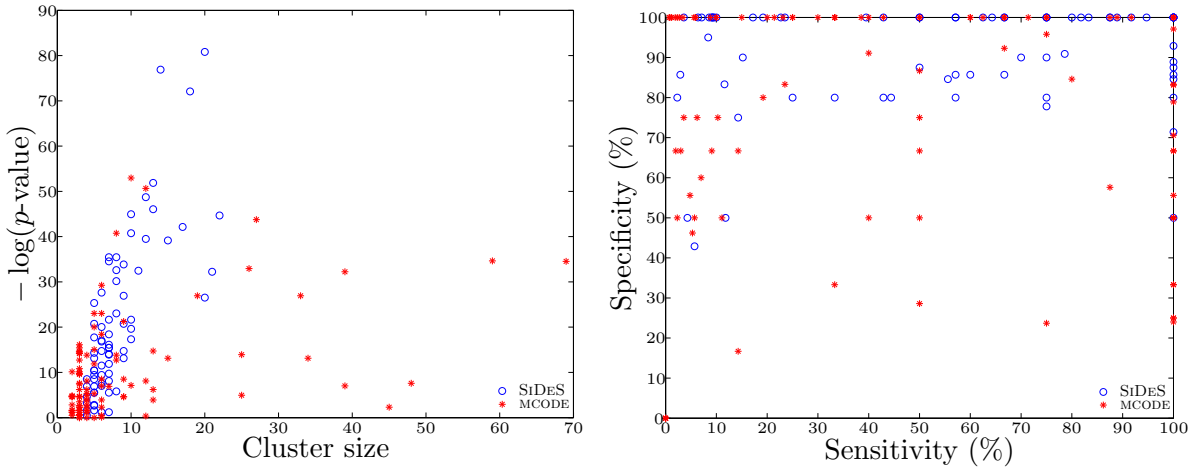


Fig. 5.6. The behavior of largest dense subgraph size and largest conserved dense subgraph size with respect to density threshold for *S. cerevisiae* and *H. sapiens* PPI networks.

each cluster with a score defined as the ratio of number of interactions to the number of proteins in the cluster. SIDES, on the other hand, associates each cluster with a p-value, which estimates the likelihood of observing the number of interactions between an identical number of proteins in a graph generated by the reference model, as discussed in Section 5.1.

We evaluate the biological relevance of identified clusters based on Gene Ontology [23, 167]. We estimate the statistical significance of the enrichment of each GO



(a) (b)

Fig. 5.7. Comparison of the performance of MCODE and SIDES algorithms in identifying dense clusters in yeast PPI network.

term in the cluster using Ontologizer [168]. For a given cluster, Ontologizer associates each GO term with a p-value, which estimates the probability of the observed enrichment of the GO term in a set of randomly chosen proteins conditioned on the enrichment of the parents of the term in GO hierarchy, based on a reference model that assumes hypergeometric distribution of GO terms among proteins.

The distribution of the p-value for the most significant annotation with respect to cluster size for clusters identified by SIDES and MCODE on the yeast PPI network is shown in Figure 5.7(a). Since each cluster is generally associated with more than one significant GO term, we report the most significant term in this figure, since this term corresponds to the most biologically meaningful annotation from a statistical perspective. On the *S. cerevisiae* PPI network, SIDES identifies 73 significant dense subgraphs, while MCODE discovers 103 dense clusters. As evident in the figure, SIDES tends to discover smaller clusters as compared to MCODE and preserves specificity of identified clusters in terms of GO annotations irrespective of cluster size. Furthermore, cluster size and significance of GO annotation are significantly correlated (0.76, $p < 9e - 15$) for SIDES, suggesting that SIDES is able to tune

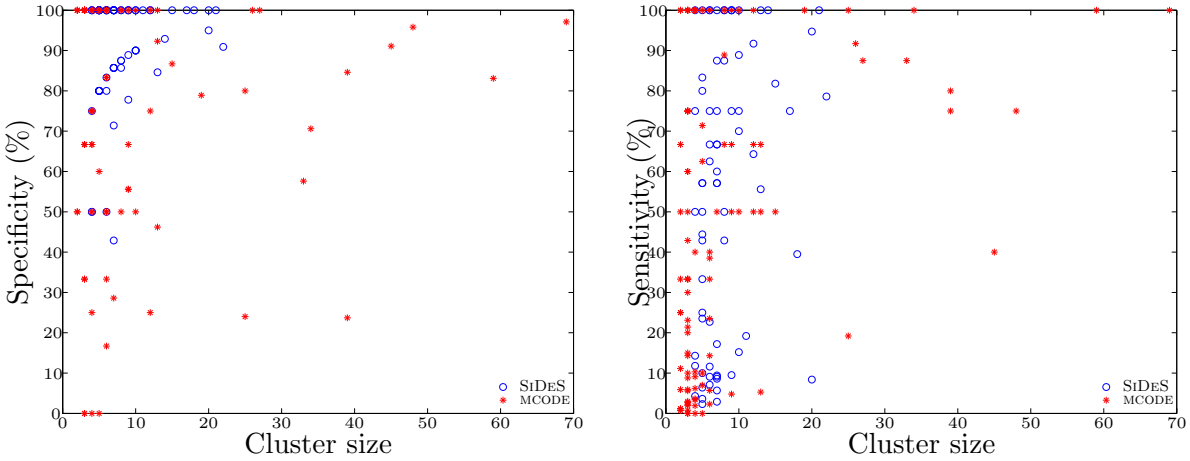


Fig. 5.8. Behavior of specificity and sensitivity with respect to cluster size for dense clusters identified by the SiDeS and MCODE algorithms.

the size of cluster to accurately capture the "meaning". The correlation of size and significance for MCODE is 0.43 ($p < 5e-06$).

In order to quantify the quality of the clusters with respect to GO annotations, we use two metrics measuring the *specificity* and *sensitivity* of a cluster with respect to the associated GO term. Assume that a cluster C containing n_C proteins is associated with a term T that is attached to n_T proteins in the set of all proteins in the network. Then, if n_{CT} of the proteins in C are attached to T, we define specificity as

$$specificity = 100 \times \frac{n_{CT}}{n_C}, \tag{5.29}$$

measuring the purity of the cluster with respect to the corresponding term. Similarly, sensitivity is defined as

$$sensitivity = 100 \times \frac{n_{CT}}{n_T}, \tag{5.30}$$

measuring the extent to which the cluster represents the corresponding term.

Since a single cluster is generally associated with more than one significant annotation, we define the specificity and sensitivity of a cluster as the maximum among all significant annotations. In other words, specificity of a cluster measures the functional

Table 5.1
Comparison of SIDES and MCODE algorithms in terms of their specificity and sensitivity with respect to GO annotations.

	SIDES			MCODE		
	Min.	Max.	Avg.	Min.	Max.	Avg.
Specificity (%)	43.0	100.0	91.2	0.0	100.0	77.8
Sensitivity (%)	2.0	100.0	55.8	0.0	100.0	47.6

purity of a cluster, while sensitivity measures the ability of the cluster to represent a functional annotation alone. The scatter-plot of specificity vs. sensitivity for all clusters discovered by the two algorithms is shown in Figure 5.7(b). As evident in the figure, only three of the 73 SIDES clusters have specificity less than 70%. Most (62%) of the circles (corresponding to SIDES clusters) reside on the upper right quarter of the plane, illustrating SIDES's ability to accurately identify most of the proteins taking part in a specific process, while maintaining specificity of the enrichment of clusters. The behavior of cluster specificity and sensitivity with respect to cluster size is shown in Figure 5.8. Correlation of size and specificity for SIDES is 0.22 ($p < 0.06$), while it is -0.02 ($p < 0.83$) for MCODE. Note that if the clusters were constructed randomly, size and specificity would be negatively correlated. The positive correlation for SIDES's clusters is illustrative of SIDES's ability of tuning cluster size to optimize specificity. Correlation of size and sensitivity for SIDES is 0.27 ($p < 0.02$), while it is 0.36 ($p < 2e - 04$) for MCODE. If the clusters were constructed at random, one would expect strong positive correlation between size and sensitivity.

A comparison of clusters identified by SIDES and MCODE in terms of biological specificity and sensitivity is shown on Table 5.1. As seen in the table, SIDES is about 20% more specific and 15% more sensitive than MCODE on the yeast network on average. As would be expected, this significant increase in accuracy comes at the price of increased computation time. In other words, MCODE is faster than SIDES

since it adapts a greedy heuristic with local optimization, while SIDES solves a more expensive min-cut algorithm repeatedly and the resulting recursion tree is generally imbalanced. For a cluster, zero specificity or sensitivity corresponds to the case where no significant annotation for the cluster is found. Note that, for all of the 73 SIDES clusters, at least one GO term is significantly enriched in the cluster.

The most significant dense subgraphs identified by SIDES in the yeast PPI network are shown in Table 5.2. As seen in the table, SIDES is able to capture many protein complexes, including transcription factor complex, mRNA cleavage factor complex, proteosome complex, nuclear ubiquitin ligase complex, mediator complex, schistoseome complex, exosome, oligosaccharyl transferase complex, TRAPP complex, eukaryotic transcription initiation factor 2B complex, hydrogen-translocating V-type ATPase complex, CCR4-NOT complex, HOPS complex, and transcription export complex. The modularity of many fundamental processes is also captured by SIDES. For example, 12 nuclear ubiquitin ligase complex proteins that induce a subgraph of 62 interactions make up 91.7% of the proteins that take part in cyclin metabolism.

Significant dense subgraphs that are conserved in *S. cerevisiae* and *H. sapiens* PPI networks are shown in Table 5.3. Most of these dense components are involved in fundamental processes and the proteins that are parts of these components share a particular function. Among these, the 7-protein conserved subnet that consists of 6 Exosomal 3'-5' exoribonuclease complex subunits and Succinate dehydrogenase is interesting. As in the case of dense subgraphs in a single network, the conserved dense subgraphs provide an insight into the crosstalk between proteins that perform different functions. For example, the largest conserved subnet of 11 proteins contains Mismatch repair proteins, Replication factor C subunits, and RNA polymerase II transcription initiation/nucleotide excision repair factor TFIIH subunits, which are all involved in DNA repair. The conserved subnets identified by SIDES are small and appear to be partial, since we employ a strict interpretation of conserved interaction here. In particular, limiting the ortholog assignments to proteins that have a COG assignment

Table 5.2
Sample protein clusters that induce significant dense subgraphs on the *S. cerevisiae* PPI network and their annotation.

Size (#P, #I)	Density *p*-value	Annotation	% Spec.	% Sens.	Annot. *p*-value
(22, 145)	2e-234	[F] transcription regulator activity	90.9	6.9	4e-20
		[C] transcription factor complex	90.9	17.1	6e-20
		[P] protein amino acid acylation	63.6	32.6	1e-11
(20, 112)	4e-163	[C] mRNA cleavage factor complex	90.0	94.7	8e-36
		[P] RNA 3'-end processing	80.0	69.6	2e-16
(18, 94)	5e-138	[C] proteasome complex	94.4	39.5	5e-32
		[P] proteolysis	94.4	11.0	3e-10
		[F] peptidase activity	83.3	15.5	1e-09
(12, 62)	2e-134	[C] nuclear ubiquitin ligase complex	100.0	47.8	2e-20
		[P] cyclin catabolism	100.0	91.7	2e-14
		[F] ligase activity	90.9	9.9	8e-11
(15, 64)	6e-85	[C] spliceosome complex	93.3	18.9	1e-17
		[F] binding	100.0	1.7	2e-09
		[P] mRNA processing	100.0	11.8	1e-05
(14, 55)	5e-69	[C] exosome (RNase complex)	92.9	100.0	4e-34
		[P] mRNA catabolism	92.9	25.5	2e-06
(10, 38)	1e-66	[C] oligosaccharyl transferase complex	100.0	88.9	2e-18
		[P] glycoprotein metabolism	100.0	15.1	9e-09
		[F] oligosaccharyl transferase activity	100.0	88.9	3e-07
(8, 20)	2e-21	[C] HOPS complex	75.0	100.0	8e-14
		[P] vacuole organization & biogenesis	75.0	17.6	1e-07
(7, 15)	3e-13	[C] exocyst	100.0	87.5	4e-16
		[P] exocytosis	100.0	20.0	1e-05

Table 5.3
Most significant conserved dense subgraphs in *S. cerevisiae* and *H. sapiens* PPI networks and their functional enrichment according to COG functional annotations.

# Prot.	# Cons. Int.	p-value	COG Annotation
10	17	10^{-68}	RNA polymerase (100%)
11	11	10^{-26}	Mismatch repair (33%)
			RNA polym. II TI/nucleotide excision rep. fac. TFIIH (33%)
			Replication factor C (22%),
7	7	10^{-25}	Exosomal 3'-5' exoribonuclease complex (86%)
4	4	10^{-24}	Single-stranded DNA-binding replication protein A (50%)
			DNA repair protein (50%)
5	4	10^{-12}	Small nuclear ribonucleoprotein(80%)
			snRNP component (20%)
5	4	10^{-12}	Histone (40%)
			Histone transcription regulator (20%)
			Histone chaperone (20%)
3	3	10^{-9}	Vacuolar sorting protein (33%)
			RNA polym. II transc. fac. comp. subunit (33%)

and considering only matching direct interactions as conserved interactions, limits the ability of the algorithm to identify a comprehensive set of conserved dense graphs. Algorithms that rely on sequence alignment scores and consider indirect or probable interactions [19,81,82] coupled with adaptation of the statistical framework presented in this paper have the potential of increasing the coverage of identified patterns, while correctly evaluating the interestingness of observed patterns.

6. CONCLUDING REMARKS AND AVENUES FOR FUTURE RESEARCH

In this dissertation, we propose efficient algorithms for identification of conserved network patterns through comparative analysis of molecular interaction networks, coupled with detailed statistical models and analyses for assessing the significance of such patterns. Proposed algorithms illustrate the importance of incorporating domain-specific semantic information in design of algorithms to provide reliable computational performance and facilitate real-time analysis. It is important to note that the tools presented here are publicly available and find widespread application in various areas of research in molecular biology. This is illustrative of the need for fast, reliable, and accessible computational network analysis tools, as more interaction data becomes available.

An important point relating to the accessibility of these tools is the necessity of elegant visualization tools and user interfaces to accompany such algorithmic approaches. As interaction data is closely related to other sources of biological data, which is also commonly needed in network analysis, standardization of data representation and storage is another important component in network analysis. Development of such infrastructure is already in progress in terms of semantic web technologies and portable visualization applications and libraries, as well as various knowledge bases. Efficient algorithmic techniques described in this dissertation, coupled with such efforts, has the potential of being established as the basic tool for daily research in systems biology, as is the case for BLAST in comparative and functional genomics.

The ability of the proposed algorithms to discover conserved substructures in molecular interaction networks, and the diversity of these patterns also encourage more detailed phylogenetic network analysis. Various research questions arise from the investigation of the results presented in this dissertation. These questions include

frequent subgraph discovery in a phylogenetic setting, phylogenetic analysis of computationally identified modules, and regression of evolutionary models to adjust and tune parameters for network alignment. With the rapid increase in quality and quantity of interaction data, algorithms presented here as well as those trying to answer these questions are critical.

LIST OF REFERENCES

LIST OF REFERENCES

[1] S. F. Altschul, T. L. Madden, A. A. Schffer, Z. Z. J. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: A new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, 1997.

[2] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "CLUSTAL-W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic Acids Research*, vol. 22, pp. 4673–4680, 1994.

[3] L. H. Hartwell, J. J. Hopfield, S. Leibler, and A. W. Murray, "From molecular to modular cell biology," *Nature*, vol. 402, pp. C47–C51, 1999.

[4] 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*, vol. 415, no. 6868, pp. 141–147, 2002.

[5] Y. Ho, A. Gruhler, A. Heilbut, G. Bader, L. Moore, and S. Adams, "Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry," *Nature*, vol. 415, pp. 180–183, 2002.

[6] 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*, vol. 98, no. 8, pp. 4569–4574, 2001.

[7] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, and J. M. Rothberg, "A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae," *Nature*, vol. 403, no. 6770, pp. 623–627, 2000.

[8] I. Lee, S. V. Date, A. T. Adai, and E. M. Marcotte, "A probabilistic functional network of yeast genes," *Science*, vol. 306, no. 5701, pp. 1555–1558, 2004.

[9] A. H. Tong, B. Drees, G. Nardelli, G. D. Bader, B. Brannetti, L. Castagnoli, M. Evangelista, S. Ferracuti, B. Nelson, S. Paoluzi, M. Quondam, A. Zucconi, C. W. Hogue, S. Fields, C. Boone, and G. Cesareni, "A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules," *Science*, vol. 295, pp. 321–324, 2002.

[10] G. D. Bader, I. Donalson, C. Wolting, B. F. Quellette, T. Pawson, and C. W. Hogue, "BIND-The biomolecular interaction network database," *Nucleic Acids Research*, vol. 29, no. 1, pp. 242–245, 2001.

[11] H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, and M. Kanehisa, "KEGG: Kyoto encyclopedia of genes and genomes," *Nucleic Acids Research*, vol. 27, pp. 29–34, 1999.

[12] 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*, vol. 30, pp. 303–305, 2002.

[13] F. Olken, "Biopathways and protein interaction databases." A lecture in *Bioinformatics Tools for Comparative Genomics: A Short Course*, May 2003.

[14] Z. Szallasi, "Genetic network analysis - from the lab bench to computers and back." Tutorial in *7th International Conference on Intelligent Systems for Molecular Biology (ISMB'99)*, 1999.

[15] A. Vespignani, "Evolution thinks modular," *Nature Genetics*, vol. 35, no. 2, pp. 118–119, 2003.

[16] M. Koyutürk, A. Grama, and W. Szpankowski, "An efficient algorithm for detecting frequent subgraphs in biological networks," *Bioinformatics Supplement on the 12th International Conference on Intelligent Systems for Molecular Biology (ISMB'04)*, pp. i200–i207, 2004.

[17] M. Koyutürk, Y. Kim, S. Subramaniam, W. Szpankowski, and A. Grama, "Detecting conserved interaction patterns in biological networks," *Journal of Computational Biology*, in press.

[18] 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)*, vol. LNCS 3500, pp. 48–65, 2005.

[19] M. Koyutürk, Y. Kim, U. Topkara, S. Subramaniam, W. Szpankowski, and A. Grama, "Pairwise alignment of protein interaction networks," *Journal of Computational Biology*, vol. 13, no. 2, pp. 182–189, 2006.

[20] R. Sharan and T. Ideker, "Modeling cellular machinery through biological network comparison," *Nature Biotechnology*, vol. 24, no. 4, pp. 427–433, 2006.

[21] M. Koyutürk, A. Grama, and W. Szpankowski, "Assessing significance of connectivity and conservation in protein interaction networks," in *Proceedings of 10th International Conference on Research in Computational Molecular Biology (RECOMB'06)*, vol. LNBI 3909, pp. 45–59, 2006.

[22] M. Koyutürk, A. Grama, and W. Szpankowski, "Assessing significance of connectivity and conservation in protein interaction networks," *Journal of Computational Biology*, submitted.

[23] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock, "Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[24] Z. N. Oltvai and A. L. Barabási, "Life's complexity pyramid," *Science*, vol. 298, pp. 763–764, 2002.

[25] H. Kitano, "Systems biology: A brief overview," *Science*, vol. 295, no. 5560, pp. 1662–1664, 2002.

[26] B. Titz, M. Schlesner, and P. Uetz, "What do we learn from high-throughput protein interaction data?," *Expert Review of Proteomics*, vol. 1, no. 1, pp. 111–121, 2004.

[27] H.-W. Mewes, C. Amid, R. Arnold, D. Frishman, U. Güldener, G. Mannhaupt, M. Musterkötter, P. Pagel, N. Strack, V. Stümpflen, J. Warfsmann, and A. Ruepp, "MIPS: Analysis and annotation of proteins from whole genomes," *Nucleic Acids Research*, vol. 32, pp. 41–44, 2004.

[28] P. Pagel, S. Kovac, M. Oesterheld, B. Brauner, I. Dunger-Kaltenbach, G. Frishman, C. Montrone, P. Mark, V. Stümpflen, H.-W. Mewes, A. Ruepp, and D. Frishman, "The MIPS mammalian protein-protein interaction database," *Bioinformatics*, vol. 21, no. 6, pp. 832–834, 2005.

[29] A. Zanzoni, L. Montecchi-Palazzi, M. Quondam, G. Ausiello, M. Helmer-Citterich, and G. Cesareni, "MINT: A molecular interaction database," *FEBS Letters*, vol. 513, no. 1, pp. 135–140, 2002.

[30] D. Scholtens, M. Vidal, and R. Gentleman, "Local modeling of global interactome networks," *Bioinformatics*, vol. 21, no. 17, pp. 3548–3557, 2005.

[31] Y. Kim, M. Koyutürk, U. Topkara, A. Grama, and S. Subramaniam, "Inferring functional information from domain co-evolution," *Bioinformatics*, vol. 22, no. 1, pp. 40–49, 2006.

[32] 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*, vol. 96, no. 8, pp. 4285–4288, 1999.

[33] R. Jansen, D. Greenbaum, and M. Gerstein, "Relating whole-genome expression data with protein-protein interactions," *Genome Research*, vol. 12, no. 1, pp. 37–46, 2002.

[34] J. M. Stuart, E. Segal, D. Koller, and S. K. Kim, "A gene-coexpression network for global discovery of conserved genetic modules," *Science*, vol. 302, no. 5643, pp. 249–255, 2003.

[35] S. Ashtana, O. D. King, F. D. Gibbons, and F. P. Roth, "Predicting protein complex membership using probabilistic network reliability," *Genome Research*, vol. 14, pp. 1170–1175, 2004.

[36] M. A. Gilchrist, L. A. Salter, and A. Wagner, "A statistical framework for combining and interpreting proteomic datasets," *Bioinformatics*, vol. 20, no. 5, pp. 689–700, 2003.

[37] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein, "A bayesian networks approach for predicting protein-protein interactions from genomic data," *Science*, vol. 302, pp. 449–453, 2003.

[38] T. Kato, K. Tsuda, and K. Asai, "Selective integration of multiple biological data for supervised network inference," *Bioinformatics*, vol. 21, no. 10, pp. 2488–95, 2005.

[39] 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)*, pp. 203–211, 1994.

[40] 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*, vol. 32, no. 1, pp. 438–442, 2004.

[41] 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*, vol. 24, no. 1, pp. 26–28, 1996.

[42] 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*, vol. 19, no. 8, pp. 930–937, 2003.

[43] 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*, pp. 175–186, 1997.

[44] J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins, "Computational studies of gene regulatory networks: In numero molecular biology," *Nature Reviews Genetics*, vol. 2, pp. 268–279, 2001.

[45] 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*, vol. 298, no. 5594, pp. 799–804, 2002.

[46] K. Struhl, "Yeast transcriptional regulatory mechanisms," *Annual Reviews of Genetics*, vol. 29, pp. 651–674, 1995.

[47] A. G. Hinnebusch, "Evidence for translational regulation of the activator of general amino acid control in yeast," *PNAS*, vol. 81, no. 20, pp. 6442–6446, 1984.

[48] A. Bevilacqua, M. C. Ceriani, S. Capaccioli, and A. Nicolin, "Post-transcriptional regulation of gene expression by degradation of messenger RNAs," *Journal of Cellular Physiology*, vol. 195, no. 3, pp. 356–372, 2003.

[49] P. D'haeseleer, S. Liang, and R. Somogyi, "Genetic network inference: from co-expression clustering to reverse engineering," *Bioinformatics*, vol. 16, no. 8, pp. 707–726, 2000.

[50] T. S. Gardner, D. di Bernardo, D. Lorenz, and J. J. Collins, "Inferring genetic networks and identifying compound mode of action via expression profiling," *Science*, vol. 301, no. 5629, pp. 102–105, 2003.

[51] M. J. de Hoon, S. Imoto, K. Kobayashi, N. Ogasawara, and S. Miyano, "Inferring gene regulatory networks from time-ordered gene expression data of bacillus subtilis using differential equations," in *Proceedings of Pacific Symposium on Biocomputing*, vol. 8, pp. 17–28, 2003.

[52] E. Wingender, X. Chen, R. Hehl, H. Karas, I. Liebich, V. Matys, T. Meinhardt, M. Prüß, I. Reuter, and F. Schacherer, "TRANSFAC: An integrated system for gene expression regulation," *Nucleic Acids Research*, vol. 28, no. 1, pp. 316–319, 2000.

[53] W. Wang, J. M. Cherry, D. Botstein, and H. Li, "A systematic approach to reconstructing transcription networks in saccharomyces cerevisiae," *PNAS*, vol. 99, no. 26, pp. 16893–16898, 2002.

[54] U. S. Bhalla and R. Iyengar, "Emergent properties of networks of biological signaling pathways," *Science*, vol. 283, no. 5400, pp. 381–387, 1999.

[55] J. Li, Y. Ning, W. Hedley, B. Saunders, Y. Chen, N. Tindill, T. Hannay, and S. Subramaniam, "The molecule pages database," *Nature*, vol. 420, pp. 716–717, 2002.

[56] N. Pržulj, "Graph theory analysis of protein-protein interactions," in *Knowledge Discovery in Proteomics* (I. Jurisica and D. Wigle, eds.), CRC Press, 2004.

[57] N. Pržulj, D. A. Wigle, and I. Jurisica, "Functional topology in a network of protein interactions," *Bioinformatics*, vol. 20, no. 3, pp. 340–348, 2004.

[58] A. W. Rives and T. Galitski, "Modular organization of cellular networks," *PNAS*, vol. 100, no. 3, pp. 1128–1133, 2003.

[59] H. Yu, X. Zhu, D. Greenbaum, J. Karro, and M. Gerstein, "TopNet: A tool for comparing biological sub-networks, correlating protein properties with topological statistics," *Nucleic Acids Research*, vol. 32, no. 1, pp. 328–337, 2004.

[60] A. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[61] H. Jeong, S. P. Mason, A. Barabási, and Z. N. Oltvai, "Lethality and centrality in protein networks," *Nature*, vol. 411, pp. 41–42, 2001.

[62] S. Wuchty, "Evolution and topology in the yeast protein interaction network," *Genome Research*, vol. 14, pp. 1310–1314, 2004.

[63] G. D. Bader and C. W. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," *BMC Bioinformatics*, vol. 4, no. 2, 2003.

[64] V. Spirin and L. A. Mirny, "Protein complexes and functional modules in molecular networks," *PNAS*, vol. 100, no. 21, pp. 12123–12128, 2003.

[65] J. B. Pereira-Leal, A. J. Enright, and C. A. Ouzounis, "Detection of functional modules from protein interaction networks," *Proteins*, vol. 54, no. 1, pp. 49–57, 2004.

[66] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A. L. Barabási, "Hierarchical organization of modularity in metabolic networks," *Science*, vol. 297, pp. 1551–1555, 2002.

[67] J. Gagneur, R. Krause, T. Bouwmeester, and G. Casari, "Modular decomposition of protein-protein interaction networks," *Genome Biology*, vol. 5, p. R7, 2004.

[68] H. Ma, X. Zhao, Y. Yuan, and A. Zeng, "Decomposition of metabolic network into functional modules based on the global connectivity structure of reaction graph," *Bioinformatics*, vol. 20, no. 12, pp. 1870–6, 2004.

[69] J. Berg and M. Lassig, "Local graph alignment and motif search in biological networks," *PNAS*, vol. 101, no. 41, pp. 14689–14694, 2004.

[70] 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*, vol. 101, no. 16, pp. 5934–5939, 2004.

[71] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, "Network motifs in the transcriptional regulation network of Escherichia coli," *Nature Genetics*, vol. 31, no. 1, pp. 64–68, 2002.

[72] S. Wuchty, Z. N. Oltvai, and A. L. Barabási, "Evolutionary conservation of motif constituents in the yeast protein interaction network," *Nature Genetics*, vol. 35, no. 2, pp. 176–179, 2003.

[73] J. S. Bader, "Greedily building protein networks with confidence," *Bioinformatics*, vol. 19, pp. 1869–1874, 2003.

[74] C. Brun, C. Herrmann, and A. Guénoche, "Clustering proteins from interaction networks for the prediction of cellular functions," *BMC Bioinformatics*, vol. 5, no. 95, 2004.

[75] S. Letovsky and S. 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)*, pp. 197–204, 2003.

[76] J.-D. J. Han, N. Bertin, T. Hao, D. S. Goldberg, G. F. Berriz, L. V. Z̃hang, D. Dupuy, A. J. M. Walhout, M. E. Cusick, F. P. Roth, and M. Vidal "Evidence for dynamically organized modularity in the yeast protein interaction network," *Nature*, vol. 430, pp. 88–93, 2004.

[77] M. R. Said, T. J. Begley, A. V. Oppenheim, D. A. Lauffenburger, and L. D. Samson, "Global network analysis of phenotypic effects: Protein networks and toxicity modulation in Saccharomyces cerevisiae," *PNAS*, vol. 101, no. 52, pp. 18006–18011, 2004.

[78] H. Yu, D. Greenbaum, H. X. Lu, X. Zhu, and M. Gerstein, "Genomic analysis of essentiality within protein networks," *Trends in Genetics*, vol. 20, no. 6, pp. 227–231, 2004.

[79] M. Medina, "Genomes, phylogeny, and evolutionary systems biology," Tech. Rep. LBNL-57355, Lawrence Berkeley National Laboratory, 2005.

[80] T. Dandekar, S. Schuster, B. Snel, M. Huynen, and P. Bork, "Pathway alignment: application to the comparative analysis of glycolytic enzymes," *Biochemical Journal*, vol. 343, pp. 115–124, 1999.

[81] 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 8th International Conference on Research in Computational Molecular Biology (RECOMB'04)*, pp. 282–289, 2004.

[82] 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*, vol. 102, no. 6, pp. 1974–1979, 2005.

[83] Y. Tohsato, H. Matsuda, and A. Hashimoto, "A multiple alignment algorithm for metabolic pathway analysis using enzyme hierarchy," in *Proceedings of 8th International Conference on Intelligent Systems for Molecular Biology (ISMB'00)*, pp. 376–383, 2000.

[84] H. Hu, X. Yan, Y. Huang, J. Han, and X. J. Zhou, "Mining coherent dense subgraphs across massive biological networks for functional discovery," *Bioinformatics*, vol. 21, no. Suppl. 1, pp. i213–i221, 2005.

[85] 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*, vol. 32, pp. W83–W88, 2004.

[86] A. V. Antonov, I. V. Tetko, and H. W. Mewes, "A systematic approach to infer biological relevance and biases of gene network structures," *Nucleic Acids Research*, vol. 34, no. 1, p. e6, 2006.

[87] S. Itzkovitz, R. Milo, N. Kashtan, G. Ziv, and U. Alon, "Subgraphs in random networks," *Physical Review E*, vol. 68, no. 026127, 2003.

[88] M. Deng, S. Mehta, F. Sun, and T. Chen, "Inferring domain-domain interactions from protein-protein interactions," *Genome Research*, vol. 12, no. 10, pp. 1540–1548, 2002.

[89] H. Lee, M. Deng, F. Sun, and T. Chen, "An integrated approach to the prediction of domain-domain interactions," *BMC Bioinformatics*, vol. 7, no. 269, 2006.

[90] S. A. Teichmann, A. G. Murzin, and C. Chothia, "Determination of protein function, evolution and interactions by structural genomics," *Current Opinions in Structural Biology*, vol. 11, no. 3, pp. 354–363, 2001.

[91] P. Bertone and M. Gerstein, "Integrative data mining: The new direction in bioinformatics," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 4, pp. 33–40, 2001.

[92] S. Tornow and H. W. Mewes, "Functional modules by relating protein interaction networks and gene expression," *Nucleic Acids Research*, vol. 31, no. 21, pp. 6283–6289, 2003.

[93] H. Wu, Z. Su, F. Mao, V. Olman, and Y. Xu, "Prediction of functional modules based on comparative genome analysis and gene ontology application," *Nucleic Acids Research*, vol. 33, no. 9, pp. 2822–2837, 2005.

[94] E. M. Marcotte, M. Pellegrini, H. L. Ng, D. W. Rice, T. O. Yeates, and D. Eisenberg, "Detecting protein function and protein-protein interactions from genome sequences," *Science*, vol. 285, pp. 751–753, 1999.

[95] F. Pazos and A. Valencia, "Similarity of phylogenetic trees as indicator of protein-protein interaction," *Protein Engineering*, vol. 14, no. 9, pp. 609–614, 2001.

[96] T. Sato, Y. Yamanishi, K. Horimoto, H. Toh, and M. Kanehisa, "Prediction of protein-protein interactions from phylogenetic trees using partial correlation coefficient," *Genome Informatics*, vol. 14, pp. 496–497, 2003.

[97] A. K. Ramani and E. M. Marcotte, "Exploiting the co-evolution of interacting proteins to discover interaction specificity," *Journal of Molecular Biology*, vol. 327, pp. 273–284, 2003.

[98] Y. Kim and S. Subramaniam, "Locally defined protein phylogenetic profiles reveal previously missed protein interactions and functional relationships," *Proteins*, vol. 62, no. 4, pp. 1115–1124, 2006.

[99] J. F. Poyatos and L. D. Hurst, "How biologically relevant are interaction-based modules in protein networks?," *Genome Biology*, vol. 5, no. R93, 2004.

[100] C. von Mering, E. M. Zdobnov, S. Tsoka, F. D. Ciccarelli, J. B. Pereira-Leal, C. A. Ouzounis, and P. Bork, "Genome evolution reveals biochemical networks and functional modules," *PNAS*, vol. 100, no. 26, pp. 15428–15433, 2003.

[101] T. Yamada, S. Goto, and M. Kanehisa, "Extraction of phylogenetic network modules from prokayrote metabolic pathways," *Genome Informatics*, vol. 15, pp. 249–258, 2004.

[102] A. Bateman, L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E. L. L. Sonnhammer, D. J. Studholme, C. Yeats, and S. R. Eddy, "The Pfam protein families database," *Nucleic Acids Research*, vol. 32, pp. D138–D141, 2004.

[103] A. Heger and L. U. Holm, "Exhaustive enumeration of protein domain families," *Journal of Molecular Biology*, vol. 328, no. 3, pp. 749–767, 2003.

[104] T. Washio and H. Motoda, "State of the art of graph-based data mining," *ACM SIGKDD Explorations Newsletter*, vol. 5, no. 1, pp. 59–68, 2003.

[105] J. Hipp, U. Güntzer, and G. Nakhaeizadeh, "Algorithms for association rule mining - A general survey and comparison," *ACM SIGKDD Explorations Newsletter*, vol. 2, no. 1, pp. 58–64, 2000.

[106] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of 20th International Conference on Very Large Data Bases (VLDB'94)*, pp. 487–499, 1994.

[107] J. R. Ullman, "An algorithm for subgraph isomorphism," *Journal of the ACM*, vol. 23, no. 1, pp. 31–42, 1976.

[108] M. Kuramochi and G. Karypis, "Frequent subgraph discovery," in *IEEE International Conference on Data Mining*, pp. 313–320, 2001.

[109] X. Yan and J. Han, "CloseGraph: Mining closed frequent graph patterns," in *Proceedings of 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, pp. 286–295, 2003.

[110] D. J. Cook and L. B. Holder, "Graph-based data mining," *IEEE Intelligent Systems*, vol. 15, no. 2, pp. 32–41, 2000.

[111] A. Inokuchi, T. Washio, and H. Motoda, "An apriori-based algorithm for mining frequent substructures from graph data," in *Proceedings of 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD'00)*, pp. 13–23, 2000.

[112] X. Yan and J. Han, "gSpan: Graph-based substructure pattern mining," in *IEEE Interntional Conference on Data Mining*, pp. 721–724, 2002.

[113] J. Huan, W. Wang, D. Bandyopadhyay, J. Snoeyink, J. Prins, and A. Tropsha, "Mining spatial motifs from protein structure graphs," in *Proceedings of 8th International Conference on Research in Computational Molecular Biology (RECOMB'04)*, pp. 308–315, 2004.

[114] J. Huan, W. Wang, J. Prins, and J. Yang, "SPIN: Mining maximal frequent subgraphs from graph databases," in *Proceedings of 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, pp. 581–586, 2004.

[115] S. Nijssen and J. N. Kok, "A quickstart in frequent structure mining can make a difference," in *Proceedings of 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, 2004.

[116] S. Ghazizadeh and S. Chawathe, "Discovering frequent structures using summaries," Tech. Rep. CS-TR-4364, Computer Science Department, University of Maryland, 2001.

[117] A. Wagner, "The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes," *Molecular Biology ad Evolution*, vol. 18, no. 7, pp. 1283–1292, 2001.

[118] A. J. Enright, S. V. Dongen, and C. A. Ouzounis, "An efficient algorithm for large-scale detection of protein families," *Nucleic Acids Research*, vol. 30, no. 7, pp. 1575–1584, 2002.

[119] J. Yang and W. Wang, "Towards automatic clustering of protein sequences," in *Proceedings of IEEE Computer Society Bioinformatics Conference (CSB'02)*, pp. 175–186, 2002.

[120] M. Remm, C. E. V. Storm, and E. L. L. Sonnhammer, "Automatic clustering of orthologs and in-paralogs from pairwise species comparisons," *Journal of Molecular Biology*, vol. 314, pp. 1041–1052, 2001.

[121] R. Tatusov, N. Fedorova, J. Jackson, A. Jacobs, B. Kiryutin, and E. Koonin, "The COG database: An updated version includes eukaryotes," *BMC Bioinformatics*, vol. 4, no. 41, 2003.

[122] D. L. Wheeler, T. Barrett, D. A. Benson, S. H. Bryant, K. Canese, V. Chetvernin, D. M. Church, M. DiCuccio, R. Edgar, S. Federhen, L. Y. Geer, W. Helmberg, Y. Kapustin, D. L. Kenton, O. Khovayko, D. J. Lipman, T. L. Madden, D. R. Maglott, J. Ostell, K. D. Pruitt, G. D. Schuler, L. M. Schriml, E. Sequeira, S. T. Sherry, K. Sirotkin, A. Souvorov, G. Starchenko, T. O. Suzek, R. Tatusov, T. A. Tatusova, L. Wagner, and E. Yaschenko, "Database resources of the national center for biotechnology," *Nucleic Acids Research*, vol. 31, no. 1, pp. 28–33, 2003.

[123] C. Liébecq, ed., *Biochemical Nomenclature and Related Documents*. Portland Press, 2 ed., 1992.

[124] R. C. Agarwal, C. C. Aggarwal, and V. V. V. Prasad, "A tree projection algorithm for generation of frequent item sets," *Journal of Parallel and Distributed Computing*, vol. 61, no. 3, pp. 350–371, 2001.

[125] D. Burdick, M. Calimlim, and J. Gehrke, "MAFIA: A maximal frequent itemset algorithm for transactional databases," in *Proceedings of 17th International Conference on Data Engineering (ICDE'01)*, pp. 443–452, 2001.

[126] K. Gouda and M. J. Zaki, "Efficiently mining maximal frequent itemsets," in *IEEE International Conference on Data Mining*, pp. 163–170, 2001.

[127] S. Iuchi, "Three classes of C2H2 zinc finger proteins," *Cellular and Molecular Life Sciences*, vol. 58, pp. 625–635, 2001.

[128] R. Auty, H. Steen, L. Myers, J. Persinger, B. Bartholomew, S. Gygi, and S. Buratowski, "Purification of active TFIID from Saccharomyces cerevisiae: Extensive promoter contacts and co-activator function," *Journal of Biological Chemistry*, vol. 279, no. 48, pp. 49973–49981, 2004.

[129] E. Bouveret, G. Rigaut, A. Shevchenko, M. Wilm, and B. Seraphin, "A Sm-like protein complex that participates in mRNA degradation," *EMBO Journal*, vol. 19, no. 7, pp. 1661–1671, 2000.

[130] D. Winter, E. Choe, and R. Li, "Genetic dissection of the budding yeast Arp2/3 complex: A comparison of the in vivo and structural roles of individual subunits," *PNAS*, vol. 96, no. 13, pp. 7288–72893, 1999.

[131] A. Hierro, J. K. J. Sun and, A. S. Rusnak, G. Prag, S. D. Emr, and J. A. Hurley, "Structure of the ESCRT-II endosomal trafficking complex," *Nature*, vol. 431, pp. 221–225, 2004.

[132] W. Wickner and A. Haas, "Yeast homotypic vacuole fusion: A window on organelle trafficking mechanisms," *Annual Reviews of Biochemistry*, vol. 69, pp. 247–275, 2000.

[133] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, no. 1, pp. 195–197, 1981.

[134] 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*, vol. 100, no. 20, pp. 11394–11399, 2003.

[135] R. Y. Pinter, O. Rokhlenko, E. Yeger-Lotem, and M. Ziv-Ukelson, "Alignment of metabolic pathways," *Bioinformatics*, vol. 21, no. 16, pp. 3401–3408, 2005.

[136] F. Chung, L. Lu, T. G. Dewey, and D. J. Galas, "Duplication models for biological networks," *Journal of Computational Biology*, vol. 10, no. 5, pp. 677–687, 2003.

[137] E. Eisenberg and Y. Levanon, "Preferential attachment in the protein network evolution," *Physical Review Letters*, vol. 91, no. 13, p. 138701, 2003.

[138] R. Pastor-Satorras, E. Smith, and R. V. Solé, "Evolving protein interaction networks through gene duplication," *Journal of Theoretical Biology*, vol. 222, pp. 199–210, 2003.

[139] A. Vázquez, A. Flammini, A. Maritan, and A. Vespignani, "Modeling of protein interaction netwokrs," *ComPlexUs*, vol. 1, pp. 38–44, 2003.

[140] A. Wagner, "How the global structure of protein interaction networks evolves," *Proceedings of the Royal Society B: Biological Sciences*, vol. 270, no. 1514, pp. 457–466, 2003.

[141] J. Berg, M. Lässig, and A. Wagner, "Structure and evolution of protein interaction networks: A statistical model for link dynamics and gene duplications," *BMC Evolutionary Biology*, vol. 5, no. 51, 2004.

[142] H. B. Fraser, A. E. Hirsh, L. M. Steinmetz, C. Scharfe, and M. W. Feldman, "Evolutionary rate in the protein interaction network," *Science*, vol. 296, no. 5568, pp. 750–752, 2002.

[143] V. Kunin, J. B. Pereira-Leal, and C. A. Ouzounis, "Functional evolution of the yeast protein interaction network," *Molecular Biology ad Evolution*, vol. 21, no. 7, pp. 1171–1176, 2004.

[144] H. Qin, H. H. S. Lu, W. B. Wu, and W. Li, "Evolution of the yeast protein interaction network," *PNAS*, vol. 100, no. 22, pp. 12820–12824, 2003.

[145] U. Feige, D. Peleg, and G. Kortsarz, "The dense k-subgraph problem," *Algorithmica*, vol. 29, no. 3, pp. 410–421, 2001.

[146] R. Hassin, S. Rubinstein, and A. Tamir, "Approximation algorithms for maximum dispersion," *Operations Research Letters*, vol. 21, pp. 133–137, 1997.

[147] B. W. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *The Bell System Technical Journal*, vol. 49, no. 2, pp. 291–307, 1970.

[148] M. Groll, L. Ditzel, J. Lowe, D. Stock, M. Bochtler, H. D. Bartunik, and R. Huber, "Structure of 20S proteasome from yeast at 2.4 å resolution," *Nature*, vol. 386, no. 6624, pp. 463–471, 1997.

[149] H. Fu, N. Reis, Y. Lee, M. H. Glickman, and R. D. Vierstra, "Subunit interaction maps for the regulatory particle of the 26S proteasome and the COP9 signalosome," *The EMBO Journal*, vol. 20, no. 24, pp. 7096–7107, 2001.

[150] M. S. Cyert, "Genetic analysis of calmodulin and its targets in Saccharomyces cerevisiae," *Annual Review of Genetics*, vol. 35, pp. 647–672, 2001.

[151] M. S. Waterman and M. Vingrons, "Rapid and accurate estimates of statistical significance for sequence database searches," *PNAS*, vol. 91, pp. 4625–4628, 1994.

[152] E. Hartuv and R. Shamir, "A clustering algorithm based on graph connectivity," *Information Processing Letters*, vol. 76, pp. 171–181, 2000.

[153] S. H. Yook, Z. N. Oltvai, and A. L. Barabási, "Functional and topological characterization of protein interaction networks," *Proteomics*, vol. 4, no. 4, pp. 928–942, 2004.

[154] N. Pržulj, D. G. Corneil, and I. Jurisica, "Modeling interactome: Scale-free or geometric?," *Bioinformatics*, vol. 20, no. 18, pp. 3508–3515, 2004.

[155] A. del Sol, H. Fujihashi, and P. O'Meara, "Topology of small-world networks of protein-protein complex structures," *Bioinformatics*, vol. 21, no. 8, pp. 1311–1315, 2005.

[156] N. Pržulj, D. A. Wigle, and I. Jurisica, "Functional topology in a network of protein interactions," *Bioinformatics*, vol. 20, no. 3, pp. 340–348, 2004.

[157] J.-D. J. Han, D. Dupuy, N. Bertin, M. E. Cusick, and M. Vidal, "Effect of sampling on topology predictions of protein interaction networks," *Nature Biotechnology*, vol. 23, no. 7, pp. 839–844, 2005.

[158] A. Thomas, R. Cannings, N. A. Monk, and C. Cannings, "On the structure of protein-protein interaction networks," *Biochemical Society Transactions*, vol. 31, no. 6, pp. 1491–6, 2003.

[159] F. Chung, L. Lu, and V. Vu, "Spectra of random graphs with given expected degrees," *PNAS*, vol. 100, no. 11, pp. 6313–6318, 2003.

[160] B. Bollobás, *Random Graphs*. Cambridge, UK: Cambridge University Press, 2001.

[161] W. Szpankowski, *Average Case Analysis of Algorithms on Sequences*. New York: John Wiley & Sons, 2001.

[162] W. Aiello, F. Chung, and L. Lu, "A random graph model for power law graphs," in *Proceedings of ACM Symposium on Theory of Computing*, pp. 171–180, 2000.

[163] L. Hagen and A. B. Kahng, "New spectral methods for ratio cut partitioning and clustering," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 9, pp. 1074–85, 1992.

[164] S. Wang and J. M. Siskind, "Image segmentation with ratio cut," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 6, pp. 675–690, 2003.

[165] M. Stoer and F. Wagner, "A simple min-cut algorithm," *Journal of the ACM*, vol. 44, no. 4, pp. 585–591, 1997.

[166] P. Shannon, A. Markiel, O. Ozier, N. S. Baliga, J. T. Wang, D. Ramage, N. Amin, B. Schwikowski, and T. Ideker, "Cytoscape: A software environment for integrated models of biomolecular interaction networks," *Genome Research*, vol. 13, no. 11, pp. 2498–2504, 2003.

[167] A. Hsiao, T. Ideker, J. M. Olefsky, and S. Subramaniam, "VAMPIRE microarray suite: A web-based platform for the interpretation of gene expression data," *Nucleic Acids Research*, vol. 33, no. Web Server issue, 2005.

[168] S. Grossmann, S. Bauer, P. N. Robinson, and M. Vingron, "An improved statistic for detecting over-represented gene ontology annotations in gene sets," in *10th International Conference on Research in Computational Moecular Biology (RECOMB'06)*, pp. 85–98, 2006.

VITA

VITA

Mehmet Koyutürk received his B.S. degree in 1998 and M.S. degree in 2000 from Bilkent University, in Electrical and Electronics Engineering and Computer Engineering, respectively. He was a Ph.D. student in the Computer Science Department of Purdue University from 2001 to 2006. His research interests include bioinformatics, computational systems biology, parallel and distributed algorithms, data mining, and scientific computing.