GRAPHLETALIGN AND MCDA: COMPUTATIONAL EXPLORATION OF PROTEIN FUNCTIONS BASED ON BIOMOLECULAR INTERACTIONS

A Dissertation

by

MU-FEN HSIEH

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Dr. Sing-Hoi Sze
Committee Members,	Dr. Jianer Chen
	Dr. William Murphy
	Dr. Tiffani L. Williams
Head of Department,	Dr. Dilma Da Silva

May 2016

Major Subject: Department of Computer Science and Engineering

ABSTRACT

Proteins play an important part in cellular activities and perform essential cellular functions. Computational methods have been constantly used in the protein functional analysis. Recently, biomolecular interactions have been integrated with sequences in annotating proteins and protein functional modules because the interactions are directly associated with cellular roles of proteins. This research aims to improve computational methods for two problems: the local network alignment to identify similar functional modules across interaction networks, and the protein function prediction based on biomolecular interactions.

We introduce the algorithm GraphletAlign and argue that it is feasible to enumerate all small conserved graphlets as network alignments and include more matching proteins. Functional modules generated by a simple join of the graphlet alignments have comparable or higher sensitivity and specificity with respect to other software in terms of gene ontology (GO) functional enrichment, for human, mouse, fly, worm, yeast, *E. coli*, *H. pylori*, *S. typhimurium* and *V. cholerae.* Additionally, GraphletAlign can be applied to a single-species network.

Next, we generalize a traditional multiclass classifier, canonical discriminant analysis (CDA), to a multi-label version (MCDA) for protein function prediction, where a protein can have multiple functions. Two-step regularization is proposed to solve singularity of covariance matrices resulted by correlated functional labels and data features. The whole framework includes: calculating distances between proteins based on sequences and biomolecular interactions; transforming the distance matrices into feature vectors by applying multi-dimensional scaling (MDS); and finally transforming the features to canonical variables with MCDA which maximizes separation of functional groups for classification. We show that our framework outperforms previous multiclass binary classifiers and a multi-label classifier in predicting for *S. cerevisiae* proteins the high-level functions in the MIPS Functional Catalogues (FunCat) and the GO Slim. Furthermore, canonical variables can be projected and visualized on a two-dimensional plot, which gives an insight to the group coverage and outliers.

DEDICATION

To my parents, C. Hsieh and L. Chen

ACKNOWLEDGMENTS

I would like to thank my adviser Dr. Sing-Hoi Sze who have helped formulate my problems and methods, taught me experiment design, revised my articles for publication, and funded me for many years.

I thank my committee members Dr. Tiffani L. Williams, Dr. William Murphy and Dr. Jianer Chen who gave me interesting questions for my preliminary exams which have helped me with developing my dissertation. I thank all the supervisors of my teaching assistantship for every aspects of training.

Several individuals from the department and from other institutes have provided me with references about classification and prediction. I thank Dr. Chang-Fu Hsieh from Institute of Ecology and Evolutionary Biology, National Taiwan University, Taiwan, for personal lectures on distance matrix analysis using various similarity coefficients, principal coordinate analysis and canonical discriminant analysis. I thank Dr. Kuang-Yu Chang and Dr. Huei-Fang Yang from Institute of Information Science, Academia Sinica, Taiwan, for discussions and providing literature of multi-label classification and matrix regularization. I thank Dr. Jay Chen and Jasmine Cheng for suggestions of relevant statistical books and lecture slides. I also thank Dr. Gangman Yi for teaching me evaluation of classifications.

I extend special thanks to Dr. Teresa Leyk for encouragement throughout my PhD years.

Finally, I would like to thank all my families for constant support, especially in caring for my son in the tough times.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Proteins serve as basic working units of cell machinery. They participate in major biological functions, including basic physiology, cell regulation, flow of genetic information, cell cycle, and more. Protein function analysis is key to understanding cellular processes, and computational methodologies have been an integral part of analysis, particularly in genome-scale studies. A recent research on *Nature* 2014 (Kim et al., 2014) published the first draft map of human proteome, which identified proteins corresponding to 84% of function-annotated protein-coding genes. They proposed a computational proteogenomic analysis that incorporates peptide identification and sequence alignment in various types of databases for the purpose of not only identifying novel peptides/proteins but also correcting existing annotations. The importance of computational protein annotation has also shown on complementing the wait of manual curation of protein functions based on experimental evidences (Radivojac *et al.*, 2013).

While classic computational annotation of protein functions relied heavily on comparison of protein sequences, new types of protein data, including physical protein-protein interactions and genetic interactions, were investigated for more accurate function prediction (Pea-Castillo *et al.*, 2008; Radivojac *et al.*, 2013). Protein sequences inherited from a common ancestral protein do not necessarily imply close functions (Stein, 2001); In contrast, protein-protein interactions aim to perform cellular functions, and therefore are directly associated with protein functions. Radivojac *et al.*'s (2013) shown that by combining various structural and interaction data, new classification algorithms can outperform a baseline sequence alignment method in predicting protein functions.

Nonetheless, there is still room for improving the accuracy of protein function prediction, especially in inferring biological processes (Radivojac *et al.*, 2013). With new type of data, the interaction data, it is an ongoing investigation in how they relate mathematically to protein functions. For example, it is suggested that proteins that share physical interaction partners or are in the same pathway or complex are associated with conserved functions. Shared genetic interaction partners are also associated with conserved functions (Roguev *et al.*, 2008). Other major challenges include that a protein can have multiple functions, functions may be correlated and have a unbalanced distribution, and multiple sources of data need to be considered together in the protein function prediction.

Cellular molecules such as genes or proteins often perform a discrete function in groups, which are called functional modules (Hartwell *et al.*, 1999). Examples include protein complexes, protein pathways, and biological processes (Bader *et al.*, 2002, Pereira-Leal *et al.*, 2004). In the studies of protein functions, identifying functional modules and their functions can provide a hierarchical-level view of cellular processes. Just as in predicting individual protein functions, a combination of protein-protein interaction data and other genome-wide information can be analyzed to search for functional modules that consist of proteins and physical interactions. Analogous to sequence alignment, functional modules can be found by comparison: search of proteins of similar structures together with interactions of similar patterns that connect those proteins.

Local network alignment was developed as such a tool for cross-network comparison of local graph patterns (Kelley *et al.*, 2003; Koyutrk *et al.*, 2006; Sharan *et al.,* 2005; Flannick *et al.*, 2006). A local network alignment is constructed by matching vertices and edges across networks. Early approaches found conserved path-shape interactions, possibly corresponding to pathways, and later were extended to identify dense patterns in the networks. Conserved small seed patterns were identified and extended beyond network neighbors. The main challenge of network alignment lies in numerous possible matches between proteins and between interaction patterns which demand large computational time, memory and hardware storage. Most approaches defined, searched for, and merged high-scoring network alignments so that a smaller set of results can be derived. However, the coverage of conserved proteins and associated interactions can be improved. While network alignment was initially proposed for cross-species comparison, it is important to identify alignments of sub-networks in one species, in parallel to sequence alignment, which also identifies paralogous sequences in single species.

In this research, we introduced two computational methods to solve the problems of local network alignment for identifying protein functional modules in protein-protein interaction networks, and the problem of multi-label protein function prediction based on protein-protein interactions and genetic interactions.

1.1 Finding conserved functional modules using graphlet alignment of protein-protein interaction networks (GraphletAlign)

While most existing methods of local network alignment start with limited high-scoring seeds, we propose a computationally feasible method of local network alignment that covers as many as proteins and conserved pairs of interactions as possible, and can be applied to a network of single species or networks of multiple species. The algorithm, GraphletAlign, is based on the concept of graphlet (Prulj *et al.*, 2004), a small induced subgraph consists of two to five nodes. We define a graphlet alignment to be a set of two or more vertex-disjoint subgraphs of common topology in which homologous proteins are at the same position. Given at least one species network all placed in a graph, GraphletAlign exhaustively enumerate graphlet alignments.

We show that a simple joining of the graphlet alignments based on protein sequence similarity can create function-enriched conserved modules of protein-protein interactions. When comparing to other methods on aligning protein-protein interaction networks of *H. sapiens*, *M. musculus*, *D. melanogaster*, *C. elegans*, *S. cerevisiae*, *E. coli*, *H. pylori*, *S. typhimurium* and *V. cholerae*, the functional modules generated by GraphletAlign cover significantly more proteins in the given networks and maintain comparable or higher sensitivity and specificity with respect to Gene Ontology enrichment.

1.2 Protein function prediction from genetic and protein-protein interactions via regularized multi-label canonical discriminant analysis (MCDA)

A protein may have more than one cellular function. While most methods of protein function prediction assumed independent functional categories, recently the problem has been re-formulated as multi-label prediction considering association between functional labels. We introduce the multi-label canonical discriminant analysis (MCDA) and regularization techniques to tackle the problem of multi-label protein function prediction. Classical canonical discriminant analysis (CDA) for multiclass classification is a variant of linear discriminant analysis (LDA) that has been shown to have optimality close to Bayes classifier. We generalize CDA to handle multi-label classification, and apply two-step regularization solve singularity of MCDA covariance matrices resulted by collinearity of functional labels and data features, which produces better results than those from pseudo-inverse or single regularization.

To predict protein functions, distance matrices are created to represent protein features, from genome-wide comparisons of protein sequences, and neighborhood of genetic and protein-protein interactions. In order to apply MCDA on the protein features, a linear combination of the distance matrices is transformed into a reduced feature space by applying multi-dimensional scaling (MDS), where each protein corresponds to a variable in the space. With subsequent application of MCDA on the MDS variables, proteins are positioned at canonical space of further reduced dimensions that maximizes group separation. We show that our algorithm outperforms previous approaches in predicting for *S. cerevisiae* (Baker's yeast) proteins the categories in the MIPS Functional Catalogues (FunCat) (Mewes *et al.*, 2004) and the Gene Ontology (GO) terms in Yeast Slim. To facilitate biological investigations, protein variables can be projected and visualized on a two-dimensional plot of MCDA canonical space.

The dissertation is outlined as follows. In Chapter II, we provide a background on molecular biology including proteins, protein functions and classification schemes, and biomolecular interactions. We review the assumptions about the relationship between biomolecular interactions and protein functions. We discuss major computational approaches in studying protein functions based on biomolecular interactions. Then, we introduce the two main problems, their challenges and related work: local network alignment and identification of functional modules of proteins and interactions, and the problem of protein function prediction and multi-label prediction. We conclude the chapter with presentation of mathematical evaluation of protein functional modules and multi-label protein function prediction. In Chapter III, we present our method of local network alignment, GraphletAlign. A graphlet alignment is mathematically defined, and the branch-and-bound algorithm of enumerating graphlet alignments is presented. We describe post-processing of graphlet alignments merged into conserved modules. Alignment results of single species and up to four species are compared against six network local alignment software. In Chapter IV, we present our method of multi-label protein function prediction, the regularized multi-label canonical discriminant analysis (MCDA). First, the problem of multi-label protein function prediction is defined. The distance measures of proteins are defined based on local sequence alignment and interaction partners. Transformation of distances to multi-dimensional scaling (MDS) variables is described. Next, we review the classic canonical discriminant analysis (CDA) for multiclass classification and prediction. We introduce our multi-label version of covariance matrix estimation for MCDA, and two-step regularization for the covariance matrices. Predictions made using regularized MCDA are compared to those by a multi-label method, MDKC, and by previous protein function prediction approaches. Predictions for a combination of distance measures from sequence alignment or interactions are evaluated as well. The effects of applying regularization, prior functional label distribution and correlation on CDA are shown in prediction results. The predictions are analyzed by functional category and against MDKC. Canonical variables are also visualized by functional category on plots. We conclude with a summary of contributions and future work in Chapter VI.

2. PRELIMINARIES AND RELATED WORK

In this chapter, we provide a background on proteins, protein functions and classification schemes, biomolecular interactions, and local sequence alignment. We review applications of interaction data in exploring protein functions. Next, we introduce the problem of local network alignment and identification of functional modules of proteins and interactions, and the problem of protein function prediction. Associated challenges and related work are listed. Finally, we discuss mathematical evaluation of protein functional modules and protein function prediction.

2.1 Proteins

Proteins are the major functional molecules in a cell. The activity of a protein is determined by its structure and how it binds to other molecules. A protein has four levels of organization. The primary structure of a protein is its protein sequence, which is a linear polypeptide chain composed of a fixed-length of combination out of 20 amino acids. Each of the 20 types of amino acids has respective chemical properties and appear with similar frequencies in all proteins. A protein sequence length varies from 50 to 25000 residues while most contain 200-500 residues and the average length is approximately 250. In 70's, IUPAC define abbreviated notion of amino acids as 20 letters: $\Sigma = A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y$. An amino acid sequence is represented as a string of letters following the direction how mRNA is translated to amino acid, which is from N-terminus to C-terminus (Figure). Every amino acid sequence is identified uniquely and encodes the structure and function of the protein.

During genetic transcription (DNA to RNA) and translation (RNA to protein) described by central dogma, every three consecutive DNA nucleotides of a gene is converted into one amino acid. Nonetheless, the primary structure of a protein cannot be completely predicted from DNA sequence. The reason is that the protein structure can be altered during and after biosynthesis by mRNA alternative splicing and protein posttranslational modification. A gene can have part of the RNA encoding sequence (exons) be included or excluded (spliced) in the transcription of DNA to mRNA. Alternative splicing of mRNA sequences often results in different amino acid sequences. Protein modification further occur on amino acids by attaching phosphate, acetate, amide groups, methyl groups or other molecules, where more than 400 types of modification exist. Consequently, distinct protein isoforms are generated from one single gene and may have different functions. It is still an ongoing project to identify all protein isoforms of an organism. In the current computational studies of protein functions, although we call it "protein function prediction", the predictions are made for each gene and its child protein isoforms. Therefore, each gene and corresponding protein isoforms may be annotated with multiple functions, and this has to be taken into account when predicting protein functions.

The next level, secondary structure and tertiary structure, refer to the three-dimensional structure of a protein. They are the folded partial or whole protein sequence. The chemical properties of the three-dimensional structure determines the binding targets of a protein and hence interactions between the protein and other molecules. The highest level, quaternary structure, refers to assemblies of several protein subunits. Protein complex is an example of quaternary structure. Physical "protein-protein interactions" occur between these protein subunits (Figure).

Traditionally, biologists characterize single proteins through various experimental techniques. They identify ordered amino acids in a protein sequence through Edman Degradation and soft ionization; determine protein conformation via X-ray crystallography, or nuclear magnetic resonance (NMR) spectroscopy; locate proteins in a cell through protein fusion and fluorescence resonance energy transfer (FRET) (Alberts, 2002). With small-scale of the experiments, the studies can only focus on few proteins. Today, biologists can identify all proteins present in a cell at a certain time given controlled conditions (called proteome) (Kim, 2014) via high-throughput techniques: 2D electrophoresis, fluorescent-protein-based assays, protein expression microarray, mass spectrometry, and systematic evolution of ligands by exponential enrichment (SELEX). Large amount of experimental data requires computational analyses. Examples of computational studies include peptide spectra search for mass spectrometry sequencing (Matthiesen, 2007), protein structure prediction from sequences, and protein folding recognition.

In this research, we incorporate primary structures of proteins, the amino acid sequences, to infer protein functions. Although we do not utilize the information of secondary or tertiary structures of proteins, we use protein-protein interactions which are directly associated with protein quaternary structures and protein functions.

2.2 Protein functions

Proteins execute functions related to physiology such as metabolism and energy, and to development process, biogenesis, cell differentiation and cell fate. Proteins catalyze chemical reactions, regulate other enzyme activity, gene expression, and modify other protein sequences. Proteins participate in decoding of genetic information including transcription and translation. Proteins transport substances within and out of cell. Proteins respond to stimuli from other cells, organism and environment, for example, temperature, pH level, clean water, and air quality. Proteins also control behavior of organism. Proteins are involved in structures such as chromosome, protein complex, plasma membrane and cell wall, and also regulate protein folding. Other typical functions of proteins include but are not restricted to binding to other molecules, and pathogenesis (Slim GO). The function of a cell is dominated by the proteins within the cell (O'Connor, 2010).

Typically after the function of a protein is hypothesized, biologists sequence the protein, construct the protein structure, and explain the protein properties and functions. Proteome data obtained by the high-throughput experiments also contain the information about protein functions under a controlled environment. Computational approaches have been used to analyze protein functions, traditionally based on similarity of protein sequences or structures (Loewenstein *et al.*, 2009). A protein is hypothesized having a function if it has a similar sequence or structure to another protein annotated with that function, which is called annotation transfer. For the purpose, computational tools have been developed to search for similar protein sequences, domains, structures or phylogeny. Automated functional annotation can facilitate formation of biological hypotheses, design of experiments, and complement the wait for manual curation or experimental verification (Radivojac *et al.*, 2013).

2.3 Protein function classification schemes

Biologists have long been putting efforts in defining a common function classification scheme for genes and proteins. Among the earliest (Kanehisa, 2005) were hierarchical Enzyme Commission number (EC number) system by IUBMB and IUPAC (Barrett *et al.*, 1992), *E. coli* role categories (Riley, 1993) and TIGR role scheme for microbes when the *H. influenzae* genome was published at 1995. EC number was integrated to the KEGG categories (Kanehisa *et al.*, 2004) which mostly focused on gene orthology and molecular-level annotation such as metabolic pathway, metabolites and biochemical reactions.

To study protein functions, we use the most popular Gene Ontology (GO) (Ashburner *et al.*, 2000) in this research, specifically the annotations of "biological processes” out of the total three subsets of ontologies other than “cellular components” and “molecular functions”. The tree-like classification scheme (actually a directed acyclic graph; DAG) was proposed at 1999 under collaboration of three model organism databases: yeast, fly and mouse, and has remained independent of species despite increasing number of contributors. Typically a higher-level GO term is more general than other descendants and can be used to annotate more genes or proteins. A protein annotated with a lower-level GO term is supposed to have all the roles from parent GO terms. Most of the GO annotations were assigned manually by curators based on literature, and some were computationally generated. The annotation evidences are described by 21 evidence codes. Only the GO terms which were annotating with experimental evidences, coded by EXP, IDA, IPI, IMP, IGI, and IEP, were adopted in this research.

As there are large number of GO terms, GO Consortium and several species databases each provides a subset of terms, GO slim, for a summary view of annotations. For example, on 2014/05/17, the Yeast slim contains 101 GO terms out of the 15-level 26752 biological-process terms, such as mitotic cell cycle and protein complex biogenesis. GO slim can provide a preliminary classification and evaluation of gene functions.

Besides assigning GO terms to individual genes, one can determine if a GO term is significantly enriched in a set of genes using the tool GO TermFinder (Boyle, 2004). GO TermFinder assumes a Bonferroni-corrected hypergeometric distribution for GO term annotations. Given total n genes in which l genes are annotated by GO term g, and given a size-s gene subset in which k genes are annotated by g, GO TermFinder calculates the probability of seeing at least k g-annotated genes in k randomly drawn genes.

$$\Pr(x \geq k|g) = 1 - \sum_{i=0}^{k-1} \frac{\binom{l}{i}\binom{n-l}{s-i}}{\binom{n}{s}} \tag{2.1}$$

If the probability is lower than a threshold α, g is significantly represented in the gene subset. In this research, GO TermFinder is used to identify significantly present functions in each subset of proteins and associated protein-protein interactions.

In addition to gene ontology, the Munich Information Center for Protein Sequences (MIPS; the current IBIS Institute of Bioinformatics and Systems Biology) also proposed a strict tree-structured classification scheme FunCat for *S. cerevisiae* proteins. Later it was extended to prokaryotes, fungi and animals (Ruepp *et al.*, 2004). There were total 1307 categories at 5 levels, which is shallower than gene ontology. The first level contains 18 functional categories which can provide a broad classification of proteins like GO slim. In this research, we utilize the 17 first-level FunCat in classifying *S. cerevisiae* protein functions.

One thing to note is that current protein function annotations are actually organized by corresponding genes in most public databases. As described in section 2.1, a gene can be decoded into different isoforms of proteins and perform distinct functions. However, all protein isoforms are stored along with the corresponding gene in only one entry of the databases. Therefore, a gene/protein in current databases may be associated with multiple functions.

2.4 Local protein sequence alignment

The function of a protein may be inferred by finding another structural similar protein with known function. Similar amino acid sequences infer similar higher-level structures and biochemical functions (Alberts, 2002). Sequence alignment tools are designed to find matching sequences of a gene or a protein. A sequence alignment (Figure) is defined computationally as a match between two sequences of letters that maximizes the similarity score or minimizes the edit distance between the sequences. A local sequence alignment matches subsequences of varied lengths, in contrast to a global sequence alignment, which matches the whole sequences. In the experiments of this research, two local alignment methods were utilized: the Smith and Waterman algorithm for pairwise sequence alignment, and the fast BLAST search in a database of sequences for a match of query sequence.

The Smith and Waterman (S&W) algorithm (Smith *et al.*, 1981) is a tool of local sequence alignment which can be used to align every pair of protein sequences in a species. The S&W uses dynamic programming to find the optimal scoring alignment of substrings of two input strings. Given string v and w over a set of letters Σ, such as the 20 amino acids, and a scoring matrix ψ for any pair of aligned letters from Σ, and a pair of aligned "gap" and letter from Σ, the S&W algorithm outputs the alignment of the maximal score among alignments of all substrings of v and w. The score of an local alignment ending at $v[i]$ and $w[j]$ can be expressed by a recurrence $\rho_{i,j}$, where $\rho_{i,0} = 0$ and $\rho_{0,j} = 0$, $1 \leq i \leq |v|$, $1 \leq j \leq |w|$.

$$\rho_{i,j} = max \begin{cases} 0 \\ \rho_{i-1,j} + \psi(v[i], -) \\ \rho_{i,j-1} + \psi(-, w[j]) \\ \rho_{i-1,j-1} + \psi(v[i], w[j]) \end{cases} , 1 \leq i \leq |v|, 1 \leq j \leq |w| \qquad (2.2)$$

The local alignment of v and w can be identified by looking for the maximal score $\rho_{i,j}$. The Darwin implementation of S&W uses a list of PAM substitution scoring matrix as ψ. The sequence similarity based on the PAM matrix can be calculated by

$$\mathrm{PAM}(v,w) = 10 \times (10\,\hat{}\,Pr(\frac{v \text{ and } w \text{ have common ancestor}}{v \text{ and } w \text{ are random alignment}}) - 1) \qquad (2.3)$$

BLAST, the widely used fast sequence database search software, aligns a query sequence against a database of sequences. BLAST treats the whole database as one long sequence and uses the Aho-Corasick algorithm, which was modified by S&W and Sellers *et al.*'s (1984), to identify the statistically significant "locally maximal" segments in the database (Jones and Pevzner, 2004). BLAST rates these segments using the Altschul-Dembo-Karlin statistics and calculates the expected number of segments with scores above a given score threshold θ.

$$E(\theta) = Kmne^{-\lambda\theta} \qquad (2.4)$$

The parameter m and n are lengths of the two sequences where the segments come; K is a constant; λ is calculated from frequencies of amino acids and the scoring matrix, which can be a PAM or BLOSUM substitution scoring matrix based on query size. Usually when the expected number of segments $E(\theta) \leq 0.01$ or a lower cutoff value, we say the alignment is significant and unlikely to occur by chance.

To choose local alignments, a cutoff threshold can be set for the identity rate or similarity between the sequences, or the expected occurrences of the alignment.

Local sequence alignment can be used in parallel with biomolecular interactions to predict functional modules of proteins and interactions. It can also serve as a baseline method of predicting functions for individual proteins (Radivojac *et al.*, 2013).

2.5 Biomolecular interactions

In additional to structural properties of proteins, biomolecular interactions have emerged as new materials for protein function studies. A protein-protein interaction refers to the physical contact of two proteins. A genetic interaction refers an observation of phenotype associated with two processed genes, which can be mutation, deletion, or over-expression. Other types of biomolecular interactions include DNA-protein interaction and biochemical reaction catalyzed by enzymes, but we focus on protein-protein interactions and genetic interactions for this research.

These interactions can be organized into a network model to provide a global view of biological processes and facilitate genome-scale analysis. A protein-protein interaction network or a genetic interaction network is usually defined as an undirected graph $G(V(G), E(G))$ where each protein is represented as a **node/vertex** in $V(G)$, and each interaction is represented as an undirected **link/edge** in $E(G)$ connecting two proteins. The graph may be represented by a data structure: adjacency list $N : V(G) \to \{V(G)\}$, where each protein is mapped to a set of proteins neighboring to it in the interaction network. Examples of network studies include network property analysis, network clustering, and network alignment.

2.5.1 Protein-protein interaction

Because proteins always physically interact with other molecules and perform tasks, protein-protein interactions are highly relevant to protein functions. A protein only binds to a few specific targets. By identifying these interacting partners, the roles of the proteins at cellular processes can be elucidated. Biologists have used traditional methods (Chatr-Aryamontri *et al.*, 2015): X-ray crystallography, NMR, FRET, biochemical fractionation, far western, PCA, observation of biochemical activity to discover protein-protein interactions (PPI), protein pathways, and protein complexes. In the past two decades, the application of high-throughput techniques (Alberts, 2002): affinity purification and mass spectrometry (AP/MS), co-immunoprecipitation followed by mass spectrometry, and two-hybrid system (Y2H) has revealed thousands of interactions from over one hundred species, which are available at several databases: BIND, MINT, DIP, BioGRID, IntAct, the human protein specific database (HPRD) and I2D. The species *C. elegans*, *D. melanogaster*, *H. sapiens*, and *S. cerevisiae* currently have more complete PPI data.

2.5.2 Genetic interaction

More recently, another type of biomolecular interactions, genetic interactions, have attracted systematic studies. When two genes are mutated, deleted or over-expressed and a phenotype is observed, they are suspected to a genetic interaction and functional relevance (Mani *et al.*, 2007). Examples of genetic interactions include that a processed gene results in growth defect, lethality or rescue of a strain that is processed for another gene, that a processed gene causes enhancement or suppression of a phenotype associated with another processed gene, and that combining two different processed genes in a cell changes fitness defect or causes growth defect or lethality under a given condition (Chatr-Aryamontri *et al.*, 2015). These genetic interactions give insights to how genes regulate cellular processes together. The species *S. cerevisiae*, , and *H. sapiens* currently have more complete genetic interaction data which can be found at BioGRID database.

2.5.3 Related work

There have been many computational analyses related to protein-protein interactions. Our method of network alignment is motivated by the studies of network properties, which are reviewed below.

Network generation: The protein-protein interaction networks of the budding yeast (Uetz *et al.*, 2000), fruit-fly (Giot *et al.*, 2003), nematode worm (Li *et al.*, 2004) (Figure), and human (Rual *et al.*, 2005, Stelzl *et al.*, 2005) were among the earliest published networks. Methodologies have been developed to integrate experimentally detected and computationally predicted molecular interactions (Gerstein *et al.*, 2002; Hoffmann *et al.*, 2003; Jansen *et al.*, 2003; Troyanskaya *et al.*, 2003; Bader *et al.*, 2004; Lee *et al.*, 2004; Wong *et al.*, 2004) to generate high-confidence networks.

Network evolution: Many works also looked into questions such as how networks affect evolution, and how networks evolve. They found, at protein level, highly connected proteins are more selective (Fraser *et al.*, 2002; Wuchty *et al.*, 2004, Fraser *et al.*, 2005), and gene duplication (Pastor-Satorras, 2003) might be the driving force of network evolution. At structural level, proteins are more conserved as a pair (Pagel *et al.*, 2004; Lemos *et al.*, 2004) or as a module (synergistic selection) (Fraser *et al.*, 2005; Qin *et al.*, 2003).

Network properties: Many studies have been contributed to quantify the global properties and organization of networks, for examples, the scale-free distribution of vertex degrees (Maslov *et al.*, 2002), high-betweenness of essential yeast proteins (Joy *et al.*, 2005), comparison of closeness and centrality of *E. coli* and human metabolic networks (Ma *et al.*, 2003), and common average lengths of shortest paths in metabolic networks (Ma *et al.*, 2003, Jeong *et al.*, 2000, Bilke *et al.*, 2001), the association between a yeast phenotypic function and essentiality of corresponding proteins in terms of network distance and clustering coefficient (Said *et al.*, 2004).

Other studies investigated into local properties of networks such as overly frequent patterns of interactions in biological networks compared to randomized networks. Milo *et al.* (2002) and Shen-Orr *et al.* (2002) identified connected subnetworks containing 3-4 proteins in transcriptional interaction networks of *E. coli* and *S.cerevisiae.* They referred to the highly enriched subnetworks as “network motif”, basic computational units to define classes of networks, and argued that some of the network motifs have specific function related to gene expression (Shen-Orr *et al.*, 2002). Motifs of protein-protein interaction networks may represent evolutionary conserved topological units with specific functions (Wuchty *et al.*, 2003). Prulj *et al.* (2004) proposed another term, "graphlet", to represent small induced subgraphs in the protein-protein interaction networks. They compared the frequency of 3-5 node graphlets in the PPI networks to that in four random networks and shown that PPI networks are better represented by a geometric random graph. Our network alignment study is motivated by the graphlet search and identifies conserved graphlets.

The most emphasized computational analysis of biomolecular interactions has been the functional analysis. We separate and describe it in the following section.

2.6 Exploring protein functions based on biomolecular interactions

It has been an important topic to computationally analyze gene/protein functions based on biomolecular interactions. As mentioned in the section 2.2, automated annotation of gene/protein function and prediction of protein pathways and complexes can help make functional hypotheses, design biological experiments, and also fill the gap when waiting for experimental verification. Past researches shown that by considering other gene/protein data such as interactions, co-expression data, or phylogenetic profiles, in addition to the baseline gene/protein sequences, can contribute to good performance in protein function classification (Radivojac *et al.*, 2013).

Many approaches of protein function prediction have utilized the discovery or assumptions about gene/protein functions and the networks:

1. Physically interacting proteins (from a protein-protein interactions network) are more likely to share a protein function (Roguev *et al.*, 2008)

2. A more frequent function in the network neighborhood can be a more probable function for a gene/protein (Schwikowski *et al.,* 2000)

3. Similar genetic profiles (i.e. sets of genetic interacting partners) are associated with similar cellular functions (Roguev *et al.*, 2008). Two proteins are likely to have the same function if their interaction partners are conserved (Bandyopadhyay *et al.*, 2006).

4. Functional similarity of proteins is associated with the data similarity of the proteins. (Wang *et al.*, 2015) Some would further consider similarity between functional labels itself based on the ontology structure. (Joshi *et al.*, 2004)

Four major computational research areas that link biomolecular functions and interactions are reviewed below.

1. **Identification of functional modules in a single network; Network clustering.** Based on interaction networks and other features of proteins in single species, the goal is to find modules of proteins and interactions, and annotate each module with popular functional labels of the known proteins in the module. A functional module is defined as a separate group of interacting cell molecules that performs a discrete function (Hartwell *et al.*, 1999) such as a protein complex or pathway. To determine if a hierarchical functional label is popular among the genes/proteins in a functional module, one can use the hypergeometric test introduced in section 2.3. The problem is often solved by searching of highly weighted vertex clusters (MCODE; Bader and Hogue, 2003, altaf-Ul-Amin *et al.*, 2006), graph clustering (Spirin and Mirny, 2003) or hierarchical clustering on pairwise protein distances (Arnau *et al.*, 2005; Rives and Galitski, 2003; Goldberg and Roth, 2003; Brun *et al.*, 2003; Samanta

and Liang, 2003). Sharan *et al.* (2007) reviewed that a pairwise distance for proteins based on the network topology, which can be based on network distance, shortest path distance profile, clustering coefficient, Czekanovski–Dice dissimilarity or statistical significance of common interacting partners. The Czekanovski–Dice (also called Srenson-Dice) dissimilarity used by PRODISTIN (Brun *et al.*, 2003) also appear in our method of protein function prediction.

2. **Identification of functional modules across networks; Local network alignment.** Based on comparison of interaction networks and features of proteins across more than one network, the goal is to find similar modules of proteins and interactions, and annotate each module with popular functional labels of the known proteins in the module. Local network alignment is such a technique to "align" similar proteins along with similar patterns of protein interactions across different networks (Kelley *et al.*, 2003; Koyutrk *et al.*, 2006; Sharan *et al.,* 2005; Flannick *et al.*, 2006; Dutkowski *et al.*, 2007; Guo *et al.*, 2009). A network alignment can also be viewed as an alignment of conserved modules of proteins and interactions, or functional modules. Genes/proteins in conserved functional modules are supposed to share similar functions, and hence contribute to the protein function studies.

3. **Identification of functional orthologs across networks; Global network alignment.** Based on comparison of interaction networks and features of proteins across more than one species, the goal is to find the best protein match in each species for each protein in another species. Each matching pair of proteins is called a functional ortholog of each other, which are supposed to have the most similar cellular role in the respective species (Bandyopadhyay *et al.*, 2006). A protein with unknown function can be annotated by the functions

of its functional ortholog. The concept can also be extended to multiple species. Global network alignment is such a technique that matches whole-genome proteins across species based on the interactions surrounding the proteins. Global network alignment aims to identify the best functional counterparts for the whole-genome proteins.

4. **Protein funtion prediction based on biomolecular interactions.** This topic is more of predicting functions for each individual protein, opposed to prediction of protein modules or pairs above. Given a set of proteins and their features including interactions, in which some protein have known functional labels and others do not, the goal is to create an algorithm that utilizes the functional labels and features of the known proteins, and predicts functional labels for the unknown proteins based on their features. Initially, predictions were made based on protein-protein interactions. Later, different types of data were introduced for function prediction, including protein-protein interactions, genetic interactions, co-expression data, and phenotypes (Joshi *et al.*, 2004; Deng *et al.*, 2004; Lee *et al.*, 2006; Mostafavi *et al.*, 2008). Similarly, Kelley and Ideker (2005) were the first to predict functional modules combining protein-protein interactions and genetic interactions.

In this research, we present a method of local network alignment and protein function prediction based on biomolecular interactions. The two problems are introduced in the following sections.

2.7 Local protein network alignment

Analogous to sequence alignment, network alignment was developed as a comparative tool of proteins and interactions from biomolecular networks. Network alignment "aligns" structural similar proteins along with the incident interactions across different networks. We call a subnetwork of proteins and interactions which is aligned in a network alignment a functional module. Network alignment has been used to predict conserved functional modules across networks, transfer functional labels within conserved modules (i.e. annotate conserved modules), and investigate network evolution. Some examples of questions that can be explored using network alignment include: How can we predict protein functions by comparing neighboring proteins in the networks? How can we identify functional components such as pathway, protein complexes by comparing networks across species? How do network structures evolve?

Network alignment was originally applied to metabolic pathways (Dandekar *et al.*, 1999; Forst *et al.*, 1999; Tohsato *et al.*, 2000) which are represented as linear paths similar to sequences. Enzymes in each pathway from a separate organism are aligned if they belong to the same function class. An alignment score, representing the similarity of pathways, is measured by the similarity of protein sequences or the likeness of function classes, which was motivated by sequence alignment score.

For comparison of protein-protein interaction networks, Walhout *et al.* (2000) first introduced the concept of "interolog" which refers to an evolutionarily conserved interaction in another organism for an interaction. Specifically, given an interaction of a pair of proteins, its interolog is the interaction of its respective protein orthologs or homologs in another species (Figure). The concept was used to predict protein-protein interactions in *C. elegans* from networks of *S. cerevisiae* (Matthews *et al.*, 2001). Yu *et al.* (2004) analyzed four species *S. cerevisiae*, *C. elegans*, *D. melanogaster*, and *H. pylori*, and found annotations can be transfer from an interolog to another interolog, provided that the two pairs of protein homologs have at least 80% of sequences being identical and BLAST E-values smaller than 10^{-70}. This was one of the first research to infer protein functions based on protein-protein interaction networks.

The concept of “interolog” was further extended to larger conserved patterns across species, consisting of many interolog (Figure). Kelley *et al.* (2003) first proposed to align pathways of protein-protein interaction networks of yeast and bacteria. Evaluation of alignments confirmed that conserved pathways correspond to specific functions. Kelley *et al.* (2003) introduced mathematical notions of network alignment and define alignment graph. Given two PPI networks, $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, a network alignment is defined as a subgraph in the alignment graph $G^2(V_1 \times V_2, E_1 \times E_2)$ (Figure). In G^2, every node $v \in V_1, v' \in V_2 \in V_1 \times V_2$ represents two homologous proteins from the respective networks G_1 and G_2, and every edge $\{u \in V_1, u' \in V_2\}\{v \in V_1, v' \in V_2\} \in E_1 \times E_2$ represents the existence of conserved interactions $uv \in E_1$ and $u'v' \in E_2$ from G_1 and G_2, respectively. Network alignment graph can have further variations such as allowing insertion of deletions of nodes in one species, in parallel to the indel in sequence alignment. Such approaches have been given a name, **local network alignment**, because only subneworks, local spots of networks, appear in a network alignment.

A network alignment can be interpreted as a set of conserved subnetworks, or called conserved network modules. Given m networks $(G_1, G_2, .., G_m)$, each local network alignment of proteins and interactions can be viewed as a list of m subgraphs or a list of m conserved modules, $(H_1(V_1, E_1) \in G_1, H_2(V_2, E_2) \in G_2, .., H_m(V_m, E_m) \in G_m)$. Different local network alignments can also be combined into a list of m conserved modules. Network alignment was utilized to identify functional modules across multiple species (Koyutrk *et al.*, 2006; Sharan *et al.,* 2005; Flannick *et al.*, 2006; Dutkowski *et al.*, 2007; Guo *et al.*, 2009).

After each module is identified, the statistically significant functions of the module proteins can be picked as representatives to annotate the whole module proteins. If given functional labels $L = \{l_1, l_2, \ldots, l_{|L|}\}$ and identified modules $H = \{\{H_{11}, H_{12}.., H_{1m}\}, \{H_{21}, H_{22}.., H_{2m}\}, .., \{H_{|H|1}, H_{|H|2}.., H_{|H|m}\}\}$, the functional prediction for the modules can be expressed as a set of pairs that associates each module to zero to multiple functional labels $M = \{\ldots, \{H_{ij}, l_k\}, \ldots\}$. Applications included aligning *Plasmodium falciparum* to five organisms, inferring the functions of the *Plasmodium falciparum* proteins (Suthram *et al.*, 2005), and aligning human interaction network and phenotype network to predict disease genes (Wu *et al.*, 2009).

2.7.1 Challenges

The first challenge of the network alignment lies in the numerous possibilities of vertex matches and the edge matches. Considering a path alignment of l vertices and $l-1$ edges in a basic alignment graph $G(V_1 \times V_2, E_1 \times E_2)$ of two networks $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ without "indel" vertex or edge, the first pair of vertices has worst-case $O(V_1 \times V_2)$ possible cases, and each of the remaining $l-1$ pairs of vertices has worst-case $O(d_1^{max} \times d_2^{max})$ cases. The large output results in high computational time, memory and storage burdens.

The large output leads to the second challenge: how to pick a good network alignment. Typically, the hypothesis about a good alignment is associated with more conserved vertices or network patterns, or more reliable evidences for an interaction. Scores have been assigned to conserved network vertices and edges, and only high-scoring alignments are accounted as meaningful. In addition to cutting down search space by defining a specific network alignment, most approaches utilizes a greedy search method to find high-scoring alignments.

Since a protein may have many homologs (sequence-similar proteins) in other species, it is possible that a protein is involved in more than one network alignment. Heuristics methods have been applied to reduce the appearances of a protein in the outputted network alignments, or find non-overlapping local network alignments.

2.7.2 Related work

PathBLAST (Kelley *et al.*, 2003) finds pairwise pathway alignments which are the paths in the alignment graph as described above.

- As its scoring model, PathBLAST calculates for each node and edge in the alignment graph the probabilities of true homology based on Bayes' rule and distribution of BLAST E-values in COG smoothed by monotone regression function, and the probabilities of true interactions based on distribution of number of associated studies. An alignment is assigned a score which is the sum of node and edge probability ratios under observation and random models.

- The search of pathway alignment is then formulated as a maximum weighted path problem of the directed acyclic alignment graph. A randomized approach is used: randomly ordering the vertices in the alignment graph and removing reversed edges to make a direct acyclic graph. Total $(5L!)$ random alignment graphs are generated, where L is the path length, and L=4 is chosen. For each graph, the highest-scoring length-L paths are identified using dynamic programming in polynomial time with respect to the size of alignment graph. The time complexity of finding most weighted pathway alignments in a random acyclic alignment graph is $O(L!2^L|V|^2)$. Overlapping paths are merged into modules composed of proteins and interactions.

MaWISh (Koyutrk *et al.*, 2006) utilizes the alignment graph to find pairwise alignments of arbitrary topologies beyond paths. An edge $\{u \in V_1, u' \in V_2\}\{v \in V_1, v' \in V_2\}$ exists in the alignment graph even if only one interaction uv or $u'v'$ exists in the PPI network, and it is called an interaction mismatch.

- An evolutionary model considering interaction match, mismatch and protein duplication is used when assigning a score to an alignment graph: (edge match scores – edge mismatch penalties – node duplication penalties).

- The problem is formulated as maximum weighted induced subgraph problem, which is solved by a greedy heuristic: choosing a node pair with the most incident matching edges as a seed; extending the so-called conserved hub by adding incident matching edges that increase alignment score. An alignment of which$<$ $r\%$ overlaps with found alignments is retained. The complexity of finding seeds is $O(|V|^2)$.

NetworkBLAST (Sharan *et al.,* 2005) find conserved subnetworks across multiple species, i.e. more than two PPI networks, in the "alignment graph" similar to that in PathBLAST. NetworkBLAST looks for conserved structures close to paths or cliques. NetworkBLAST allows insertion or deletion of a protein in some species and considers both immediate neighbors and the ones in one-hop distance in the alignment graph. If there are more than two networks, NetworkBLAST also allows arbitrary distance for an aligned protein pair in one of the networks.

- For each subset of proteins from an interaction network, NetworkBLAST calculates a "likelihood ratio" which is the ratio of the probabilities of observing induced edges under alignment model and under null model. Under either model, interacting probabilities that correspond to all edges in the subset are multiplied, assuming each interaction is independent. Under the alignment model, the probability of a true interaction is a combination from three sources: the appearances in literature, Pearson correlation coefficient of their gene expression profiles, and clustering coefficient. The probability follows a logistic distribution which was fit using MIPS interactions as positive cases, random protein pairs and random observed interactions as negative cases. Under the null model, each interaction probability is calculated from a set of random graphs with the same degree sequence. This model gives more credits to dense conserved subnetworks for their low occurrences in random networks than biological networks.

- The search of high-scoring aligned subgraphs is implemented with a greedy heuristic that begins with finding high-scoring seeds in the alignment graph, including 3-node subsets of 4-node seeds of arbitrary connected topology greedily extending from “each vertex”, and also 4-node high-scoring paths. The seeds are greedily extended to 15-node in the alignment graph guided by the likelihood. Post-processing is necessary for each discovered alignment that removes other highly overlapping alignments which have $> 80\%$ overlapping proteins overall or in any species. Each protein can appear in up to 4 conserved modules. We estimate the complexity of finding seeds as $O(L!2^L|V|^m)$, $L = 4$, $L' = 11$, m is the number of networks, and d is the maximum degree in a network.

- Each alignment is assigned a p-value compared to number of alignments discovered in random networks which have the same degree sequences or number of homologs across networks, and only alignments of p-value < 0.01 are retained. Each alignment is annotated by the enriched level-4 or deeper gene ontology (GO) terms (p-value < 0.01) that annotate at least half and at least 5 of the member proteins, where Bonferroni corrected p-value is calculated assuming a hypergeometric distribution of GO term members.

Grmlin (Flannick *et al.*, 2006) identifies high-scoring pairwise alignments consists of equivalent classes of nodes, and aligns progressively over multiple networks based on a phylogenetic tree. Several parameters regarding the scores need to be learned from real data.

- Grmlin computes a score for every node pair in an "equivalent class" of aligned nodes. The pairwise node score includes a likelihood ratio of a BLAST bit score under empirical model sampled from COG and a bit score under random model sampled from random protein pairs, and also includes penalties for the events of protein insertions, deletions, duplication, and divergences in the phylogeny constructed from all the proteins in this equivalence class. Graemlin also computes a ratio score for each edge the probabilities of observing the edge weight under empirical model against the random model. The distribution of edge weights under the empirical model is described by a square matrix of node labels, where the matrix can be the Complex ESM $\mathcal{N}(0.5, 0.8^2)$, or the Pathway ESM $\mathcal{N}(0.8, 0.15^2)$, assuming equivalent classes are assigned labels.

- Grmlin uses a seed-and-extension approach. First, Grmlin finds each high-scoring seed composed of two aligned d-clusters from two PPI networks. For each vertex in the network, a d-cluster is identified which consists of this vertex and $d-1$ "nearest" nodes. A "distance" between two proteins is the logarithmic interaction confidence and is increased once the edge is identified in a d-cluster. Grmlin greedily finds each pair of d-clusters from two PPI networks that match with miss probability $< p_{\text{miss}}$ and scores $> T$, $d = 4, T = 7, p_{\text{miss}} = 0.5$. The time complexity of seed identification is $O(|V|^2(d^2 + \lambda \Pr(Z_d > T)))$. In the extension step, a seed is transformed to a pairwise network alignment, by each time, adding a highest-scoring pair of nodes from the d-cluster pair and and their neighboring nodes to the alignment, adjusting the equivalent classes and removing highly overlapping nodes.

- A pairwise network alignment AB of network A and B is extended to multiple network alignment by progressively being aligned to a third species C. Graemlin preserves unaligned partial networks A' and B', so by aligning to network C, Graemlin generated ABC, A'C, B'C and C'.

MNAligner (Li *et al.*, 2007) formulates the pairwise network alignment as an integer quadratic programming (IQP) problem of identifying optimal scoring alignment between two small and dense subnetworks. The subnetworks were first extracted using MCODE software (Bader and Hogue, 2003).

- The maximization objective function consists of protein similarities, interaction confidences, and integers representing the presence/absence of aligned interactions or aligned proteins. The protein similarity between a protein pair can be joined ortholog confidence, the probability of replacing a protein by another, BLAST sequence similarity, or EC classification similarity.

- The IQP problem is transformed to an ILP (integer linear programming) problem and solved.

CAPPI (Dutkowski *et al.*, 2009) treats the multiple network alignment as a problem of constructing a complete ancestral network with edge weights, identifying connected component in the ancestral network after removing low-scoring edges, and projecting back to conserved subnetworks across species.

- Similar to Grmlin, CAPPI assumes non-overlapping groups of homologous proteins which are constructed using TRIBE-MCL algorithm with BLAST E-values. CAPPI then constructs phylogeny tree (reconciled gene trees) over each protein homolog group and determines evolutionary events, specifically an order of duplication and speciation events. CAPPI assumes a network evolutionary model that in an ancestral network, each node corresponds to an non-overlapping protein family group. For each duplication event, a duplicated node is added to the network, the same set of interactions is copied for this duplicated node with interaction probabilities associated with total number of vertices. For each speciation event, all the interactions and non-interactions are copied to the new network with different probabilities associated with total number of interactions.

- The problem of inferring ancestral network is formulated as an inference problem in a tree-structured Bayesian network with the assumed probabilities above. The problem can be solved by message-passing algorithm.

The successor of NetworkBLAST, the NetworkBLAST-M (Kalaev *et al.*, 2009) no longer utilizes the alignment graph composed of product of vertices and edges across networks to find a high-scoring multiple network alignment, but a "layered alignment graph" to find high-scoring d-subnets of d proteins from each network. In the layered alignment graph, every sequence-similar protein pair across networks is connected by an "inter-layer edge" and topologies of inter-layer connections are considered in the search.

- NetworkBLAST-M defines the same likelihood ratio as in NetworkBLAST, but calculates one ratio for each PPI interaction not for a subnetwork. The alignment score is the sum of all likelihood ratios for all interactions.

- NetworkBLAST-M also uses a seed-and-extension approach. A seed alignment is defined to be restricted high-scoring d-subnets, $d = 2$, and each protein is at distance 1 or 2 to an already identified protein in the same network. Type I. In a network alignment, the inter-layer edges connecting cross-network proteins, the d so-called k-spines, should have an identical pattern, which can be a path or a tree, representing identical relationship of sequence homology. A recursion is applied to find such seed by adding a species at a time which resulted in time complexity $O(k^3|E_{IL}|^d 3^k)$, $|E_{IL}|$ is the maximum number of inter-layer edges between two species. Type II. In a seed alignment, each k-spine should form a path consistent with the species phylogeny tree. A recursion is applied to connect two subsets of species which has time complexity is $O(k(k|V|)^{2d}|E_{IL}|^d)$. The extension step consists of adding a k-spine at a time. Depending on the topologies of k-spine and the order to add k proteins, it may take $O(k|E_{IL}|3^k)$ for relax order, $O(k|E_{IL}|)$ for restricted order, or $O(k|E_{IL}|2^k)$ for path.

DOMAIN (Guo and Hartemink, 2009) utilizes protein domain information to align networks. DOMAIN constructs an alignment graph (APE graph) where the meanings of a node and an edge are reversed. A node in the APE graph corresponds to a pair of aligned PPIs from the two species that are mediated by at least one shared domain-domain interaction, and an edge represents the conditions that the pair of aligned PPIs can be extended, such as identical aligned proteins (alignment extension, or node duplication), allowing interaction indels in one species (edge indel), or allowing interaction insertion in both species (edge jump).

- Every node in the APE graph is assigned a probability based on domain information calculated using the EM (expectation–maximization) algorithm, and every edge in the APE graph is assigned a score, k, $1/k$, or 1, depending on the edge type: alignment extension, node duplication, edge indel, or edge jump. An alignment score is multiplication of node and edge scores.

- The search for a high-scoring alignment starts from a node in the dense area of alignment graph and expands to other nodes until the stop criteria are satisfied: having reached 15 nodes in any species, the score increase by extension or the overall alignment score being too small, or no edge indel or edge jump shorter than distance 4.

- The time complexity estimated to construct the APE graph is $O(|E|^2 + |V|)$

2.8 Protein function prediction based on biomolecular interactions

The goal of protein function prediction differs from the identification of protein functional modules above in that each individual protein with unknown function (i.e. a unknown protein) is assigned functional labels. The most probable functions are assigned to an unknown protein, given functional labels of the proteins with known functions (i.e. known proteins), and also given various structural, experimental or network data for all the proteins. Usually supervised learning algorithms that learn the labels are used to derive the probabilities and the functional labels.

A basic protein function prediction problem takes inputs: a set of sample proteins $S = \{p_1, p_2, \dots, p_n\}$, observed biological features of these proteins F, a set of functional labels $G = \{g_1, g_2, \dots, g_m\}$, and a functional label association $L : K \mapsto G = \{\dots, \{p_i, g_j\}, \dots\}$ for known proteins $K \subset S$. The output should be a set of functional label prediction $L_U : U \mapsto G = \{\dots, \{p_i, g_j\}, \dots\}$ for unknown protein $U = (S - K)$. For each protein $p \in U$, the predicted labels maximize the posterior probability distribution function (p.d.f.) over all possible functional labels (Radivojac *et al.*, 2013).

$$\hat{L_U}(p) = \arg\max_{g \in G}\{\hat{\Pr}(g|p)\}, p \in U \tag{2.5}$$

The output L_U may take the form of posterior probability $\hat{\Pr}(g|p)$. By setting a threshold $0 \leq t \leq 1$ for the posterior probability, a set of predicted labels can be chosen.

$$\hat{L_U}(p) = \{g|\hat{\Pr}(g|p) > t\}, p \in U \tag{2.6}$$

Note that in L and L_U, a protein p_i can be associated with more than one label $\{g_j\}$ because a protein can have multiple functions as described in section 2.3.

Most of the literature approached the problem as **single-label binary prediction** for a specific function g. The multiclass association between proteins and their functions (Table 2.1 (a)) is converted to a binary association between functions and positive or negative protein instances of the functions (Table 2.1 (b)) (de Carvalho and Freitas, 2009) . Specifically, the binary probabilities that a protein p is assigned a function g or not are estimated.

(a)

Protein	Functional group
p_1	g_1
p_1	g_2
p_1	g_3
p_2	g_1
p_3	g_2
p_4	g_1
p_4	g_3
p_5	g_3

(b)

Function	Member	Non-member
g_1	p_1, p_2, p_4	p_3, p_5
g_2	p_1, p_3	p_2, p_4, p_5
g_3	p_1, p_4, p_5	p_2, p_3

(c)

	g_1	g_2	g_3
p_1	1	1	1
p_2	1	0	0
p_3	0	1	0
p_4	1	0	1
p_5	0	0	1

Table 2.1: Example inputs of prediction problems. (a) Input of **multiclass prediction**: proteins and their functional groups. (b) Input of **multiclass single-label binary prediction**: functional groups and the proteins associated with them. (c) Input of **multi-label prediction**: a binary representation of the association between proteins and functional groups.

Recently, protein function prediction has been treated as a **multi-label prediction** problem (Table 2.1 (c)), opposed to the single-label binary prediction. Functions are no more predicted independently, and association between the functions is considered in the multi-label prediction. However, protein function prediction is not a traditional multi-label prediction problem, because protein features such as sequences, structures or interactions cannot be directly represented as traditional numerical or categorical vectors of size $n \times 1$, such as the pedal lengths and pedal widths used in classifying the famous Iris flower data. It has been seen the inclusion of functional correlations in the predictions. A potential method is to convert the protein data to numerical vectors, and apply traditional multi-label prediction. In this research, we convert the protein data to numerical vectors by methods of dimensional reduction and propose a new multi-label prediction method.

MouseFunc (Kim *et al.*, 2008) and CAFA (Critical Assessment of Function Annotation; Radivojac *et al.*, 2003) are the names of challenges of the protein function prediction problem, which were held starting from 2006 and 2013, and have attracted many research groups. The challenges suggested another problem formulation, where the functional labels are defined as a hierarchical structure, corresponding to the gene ontology (GO) directed acyclic graph (DAG). That is, every functional label is not a single label but rather a subgraph of the GO hierarchy including a specific GO function and all its ancestors in the GO DAG. The output contains predictions of "function subgraphs" which maximize the posterior p.d.f. over all possible subgraphs of the GO DAG. In our research, we focus on solving the basic multi-label prediction problem, and thus smaller sets of functions of equal importance are chosen as representative functions, and hierarchical functional labels are not considered.

2.8.1 Challenges

The first challenge of protein function prediction would be the hierarchical structure of protein function annotations. Around 4000 *S. cerevisiae* proteins used in our study were annotated with a large number of ˜1100 gene ontology (GO) terms distributed in different levels of the GO DAG. Many function categories have severely low number of sample proteins and cannot be learned effectively by supervised algorithms. On the other hand, many approaches have the problem that prediction accuracy for a popular high-level function is low, because probably nearly half of the proteins may belong to the same function. Therefore, some literature do not deal with the hierarchical classification directly but rather choose a subset of representing functions for the prediction problem, such as MIPS FunCat functional categories. In our research, proteins are classified into the selected 17 MIPS FunCat categories and 91 slim GO terms, each protein belongs to an average of more than 2 out of 17 and 2 out of 91 categories.

The second challenge comes from the biomolecular interactions: to define how interactions relate to function annotations/labels. An interaction involves two proteins, opposed to a function, which is assigned to an individual protein. Most approaches utilized the four assumptions mentioned in section 2.6.

The second challenge leads to the question how to formulate a classification problem and define format of inputs. As mentioned above, protein features such as sequences, structures or interactions are not like traditional classification attributes which usually are represented as numerical or categorical attribute vectors of size $n \times 1$. In protein function prediction problem, each protein feature is usually formatted as a network, or a matrix. Multiple functional labels of proteins can be represented as a list for lookup only, network vertex labels, or a matrix. In a classification method, each function can be classified separately or all together.

Additionally, integrating or selecting more than one data sources can be challenging. For example, one may combine traditional sequence data and interaction data for prediction. The integration can occur when calculating pairwise protein similarity, when computing prior data probabilities, when applying classification algorithm, or after calculating posterior probabilities.

Another computational challenge of protein function prediction lies in multiplicity of functions one gene/protein can have. The problem has an output space containing $|S| \times |G|$ probabilities. Some would formulate as an exponential output space containing combination of functional labels.

Other challenges include unbalanced distribution of functional labels and correlations between functional labels (Gibaja *et al.*, 2015).

2.8.2 Related work

Most network-based prediction algorithms base on the assumption that interacting proteins are more likely to share functions, which is called the property of "local density enrichment". Sharan *et al.* (2007) listed three major prediction approaches which we rename by local neighborhood scoring, global network optimization, and Markov random field (MRF). The methods of "local neighborhood scoring" predict a function label for an unknown protein if the function label scores the most from the network neighborhood of that protein. The methods of "global network optimization" typically optimize a mathematical objective function, or called configuration, which incorporates the neighborhood functional state of each protein. Later methods such the ones from MouseFunc (Pea-Castillo *et al.*, 2008) and CAFA challenges (Radivojac *et al.*, 2013) focus more on combining kernel functions or combining classifiers. A "kernel method" utilize a kernel function which represents similarity of proteins. Typically, each data type such as protein sequence or an interaction network is represented by a kernel. A "classifier ensemble" finds the optimized way to combine different types of classifiers, where a classifier can be designed for each function, or for each specific data type.

Local neighborhood scoring.

The neighborhood counting method (also called guilt-by-association, majority voting) (Schwikowski *et al.*, 2000) predicts for an unknown protein the top 3 frequent functions among its interacting partners.

The Chi-square neighborhood method (Hishigaki *et al.*, 2001) describes if a function is present significantly in a protein's neighborhood. The χ_l^2 value represents the difference between the expected and real number of a protein's neighbors annotated with a certain function g, and χ_l^2 is calculated for neighbors at distance n. A function g is assigned to a unknown protein if its χ_l^2 value for the protein's neighborhood is maximal.

Chua *et al.* (2006) created the functional similarity weight (FS-Weight) opposed to the Czekanowsky-Dice dissimilarity for clustering (Brun *et al.*, 2003). FS-Weight treats protein pairs differently based on number of immediate neighbors. Reliabilities of experiments, neighbors at distance 2 and function frequencies are all incorporated when estimating the likelihood of a protein having a certain function.

Global network optimization.

Vazquez *et al.* (2003) chose a function g for an unknown protein p that maximizes total number of g-function sharing between p and other interacting unknown proteins, and also number of g-annotated neighbors of p. Protein function prediction is formulated as a **minimum multiway cut** problem and solved using simulated annealing.

Karaoz *et al.* (2004) proposed to maximize the common annotation between all pairs of interacting proteins, either g-annotated or not. The objective equation to maximize involves integer values for every protein: $\sum_{(u,v)\in E} p_u p_v$, where a protein v is assigned $p_v = 1$ if annotated with g; and otherwise, $p_v = -1$. An iterative algorithm is applied to update protein values to the most common values among their neighbors until converged.

GeneFAS (Joshi *et al.*, 2004) deals with multiple networks and calculates the Bayesian probability that a pair of interacting proteins in a certain network shares a certain part of GO term hierarchy. A GO term is assigned to an unknown protein which maximizes the joint probability of sharing GO term hierarchy with each of the interaction partners of each network. GeneFAS uses the Boltzmann machine model and simulated annealing to optimize the joint probability, that is, iteratively updates the joint probability for an unknown protein until equilibrium is reached.

Nabieva *et al.* (2005) formulated an **integer programming** problem that for each function g, 1 is assigned to a vertex assuming the protein is g-annotated and 0 otherwise, and also 0 or 1 is assigned to an edge for sharing function g between the pair of connecting proteins. The assignment is to maximize the weighted sum of edge scores over all functions.

The problem is solved by AMPL and CPLEX. Nabieva *et al.* also presented another method: **network flow** simulation that for each function g, flows are originated from g-annotated vertices and spread via neighborhood. An unknown protein is assigned function g if g results in the maximal total flow passing by.

StepPLR (Hu *et al.*, 2009) minimizes a logistic regression model of probabilities that every protein is assigned a certain function. The probability incorporates the topological overlap measure (TOM; Zhang and Horvath, 2005), the weighted average of neighborhood counting at distance 1 and 2, which they called affinity vectors. The TOM scores bases on weights assigned to each incidence of function sharing between pairs of proteins. The TOM score of proteins $\{p_u, p_v\}$ incorporates the function sharing between p_u and p_v, between p_u and p_v and their shared interacting partners, and also between p_u and other proteins, or p_v and other proteins. StepPLR applies stepwise variable addition and deletion and select the most useful probability functions.

The kernel logistic regression (KLR) method (Lee *et al.*, 2006) applies a logistic regression model for each function g on the similarities between a protein and other g-annotated or non-g-annotated proteins. Generalized KLR only consider highly correlated functions based on χ_l^2 value. KLR can handle multiple data sources.

Sefer *et al.* (2011) used a formulation similar to Vazquez *et al.*'s but replace simple counting of function sharing with a "distance" between two functions. They adopted four distance measures based on the GO term DAG structure. They proposed the least squared distortion (LSD) to convert the natural distances to mathematical metric distance so that the minimum multiway cut problem and subsequent integer programming can be approximated in logarithmic time.

Markov random field (MRF).

Deng *et al.* (2003, 2004) also assumed that interacting proteins should be more probable to share a function and were the first to apply the Markov random field approach. Given the sample distribution that a protein has a certain function, a global conditional probability is calculated for every protein having a certain function conditional on the functional labeling relation between every pair of interacting proteins. The relation can be of three types: two, one, or none proteins have the certain function. The conditional probability follows a Gibbs distribution, and different types of relation are assigned different weights, which are estimated using a quasi-likelihood approach. To predict for unknown proteins, each unknown protein initially is assigned a random function, and Gibbs sampling is run iteratively until the posterior probabilities are converged.

In Letovsky and Kasif's work (2003), the probabilities that the neighbors of each protein share or do not share a particular function both follow a Binomial distribution, respectively. They applied MRF propagation algorithm to update the probabilities of a unknown protein labeled by a certain function.

Leone and Pagnani (2004) defined a Gibbs potential which counts the number of interacting function-sharing protein pairs and number of neighbors sharing a particular function. The belief propagation message-passing algorithm is applied to maximize the "free energy" based on the Gibbs potential.

Kernel method.

Lanckriet *et al.* (2004) applied semi-definite programming (SDP) to combine pairwise similarity matrices (kernel matrices) computed from different types of data, and used the support vector machine (SVM) classification algorithm on the kernel matrix and binary label matrix.

Tsuda and Noble (2005) formulated a convex optimization problem for combining multiple kernels and solved the equivalent **min-max problem** using Lagrange multipliers.

GeneMANIA (Mostafavi *et al.*, 2008) generates affinity networks consist of edges weighted by Pearson correlation coefficients of gene profiles. Different affinity networks are combined by weighted sum. A label propagation algorithm based on Gaussian random field is applied to the combined network.

Diffusion State Distance (DSD; Cao *et al.*, 2013) represents the difference of "random walk" profiles of two vertices in the protein-protein interaction network. A random walk profile of vertex p contains the expected number of times visiting another vertex p' in a random walk of k steps starting from vertex p. Four prediction methods were adapted using DSD: neighborhood counting (Schwikowski *et al.*, 2000), χ_l^2 neighborhood (Hishigaki *et al.*, 2001), multiway cut approach (Nabieva *et al.*, 2005), and network flow approach (Nabieva *et al.*, 2005).

Classifier ensemble.

Obozinski *et al.* (2008) also proposed to calculate data-specific kernels, which are converted to networks, and apply data-specific SVMs. The SVMs are combined using logistic regression to be "reconciled" with the probability distribution of hierarchical GO terms. The Kullback-Leibler projection is chosen among the logistic regression methods.

Barutcuoglu *et al.* (2006) and Guan *et al.* (2008) applied a Bayesian model to ensemble GO-term-specific SVMs. By Barutcuoglu *et al.* (2006), the output of each classifier is assumed a Gaussian distribution and the prediction results are calibrated to the distribution. By Guan *et al.* (2008), SVMs are combined based on the accuracy of each SVM and the hierarchical relationship of GO terms.

Kim *et al.* (2008) combined three classifiers nave Bayes, decision tree and boosted tree and also a global network propagation approach.

Funckenstein (Tian *et al.*, 2008) combined guilt-by-association and guilt-by-profiling methods, where the formal uses a decision tree to predict function sharing, and the latter utilizes a random forest ensemble classifier to vote for each function label. The two methods are combined using logistic regression.

COmbined simGIC (COGIC; Cozzetto *et al.*, 2013) is the only method from the CAFA challenge that uses interaction data. COGIC is an average score of Graph-Information Content (simGIC) scores that represent the overlap between the GO term predictions from seven methods, in which protein interaction is utilized in a SVM regression method called FunctionSpace.

2.8.3 Multi-label prediction

Most of the above approaches deal with each functional label separately, such as calculating the probability of a particular protein having a specific function, the probability of a protein pair sharing a specific function, or creating a separate classifier for each function. Lately, the protein function prediction problem is formulated as a multi-label classification problem, and takes into account the correlation between functions, which is supposed to exist naturally.

GeneFAS (Joshi *et al.*, 2004) and Sefer *et al.*'s (2011), which we described above, both considered the similarity or distance between functional labels. GeneFAS represents a GO term annotation as a string of numbers where the string length corresponds to the GO term depth, and the similarity between two GO terms is calculated by the difference between the two strings. Sefer *et al.*'s utilize the shortest path length between every two GO terms in the GO DAG structure and other similar distance measures.

The function–function correlated multi-label approach (FCML; Wang *et al.*, 2013) represents function labels of a protein as vertex labels in a g-function specific network, where vertex labels $\{1,-1,0\}$ correspond to a g-annotated protein, a non-g-annotated protein, or an unknown protein. A network is further represented as a design matrix, and function-function correlations are calculated as cosine product of the matrix. FCML targets to solve a **network label propagation** problem, where vertex labels are propagated to network neighbors using a kernel function: Green's function. Then, the minimization objective function is formulated using the theory of reproducing kernel Hilbert space, and further incorporates the function-function correlations. The propagation problem is solved by simulation.

The maximization of data-knowledge consistency approach (MDKC; Wang *et al.*, 2015) by the same group aims to minimize the difference between data similarity of proteins and functional label similarity of proteins. The functional label similarity is represented by the same cosine function-function correlations. The optimization problem is formulated as a **non-negativity matrix factorization** problem for symmetric matrix, and solved by an iterative algorithm.

In our research method, canonical discriminant analysis (CDA), we find that by defining and calculating functional label correlation does not improve the predictions. Nonetheless, a cross product of functional labels already exists in the equation. Regularization of the label cross product can improve the performance.

2.9 Evaluation of prediction of protein functions and protein functional modules

In this research, we apply local network alignment to find functional modules of proteins and interactions, and we annotate them with the functional labels significantly enriched in the modules. We also make multi-label functional predictions for individual proteins. The results of the two problems receive different ways of performance evaluation in the literature. However, they share some of the methodology, and some of the performance measures even share the same names, such as sensitivity and specificity, which actually have different definitions for the two problems. In this section, we compare the methodology of performance evaluation for our two functional prediction methods.

Given functional labels $G = \{g_1, g_2, \ldots, g_{|G|}\}$, the functional prediction is a set of pairs $L_U = \{\ldots, \{u_i, g_j\}, \ldots\}$ that associates an unknown protein in $U = \{u_1, u_2, ..\}$ with at least one functional label. Similarly, a predicted set of protein interaction functional modules from m networks is a set of pairs $M = \{\ldots, \{H_{ij}, g_l\}, \ldots\}$ that associates each module in $H = \{H_{11}, H_{12}.., H_{1m}, H_{21}, H_{22}.., H_{2m}, .., H_{|H|1}, H_{|H|2}.., H_{|H|m}\}$ with zero to multiple functional labels, where an annotation g_l is extracted from the known proteins in the module H_{ij} and used annotate the whole module. Examples of the two types of functional associations are presented at Table 2.2, and we discuss how to evaluate the associations in this section.

(a)

Unknown protein	Predicted function
u_1	g_1, g_2, g_3
u_2	g_1
u_3	g_2
u_4	g_1, g_3
u_5	g_3

(b)

Functional module	Predicted function
H_{11}	g_1, g_2
H_{12}	g_1
H_{13}	g_2
H_{21}	g_1, g_3
H_{22}	
H_{23}	g_3

Table 2.2: Examples of predicted functions for proteins and functional modules. (a) Unknown proteins and their predicted functional groups. (b) Functional modules of proteins and interactions and the predicted functions for the proteins in the modules.

Both types of functional associations do not have a standard solution. Unknown proteins mean that we do not know the real biological functions yet; A set of "true" interaction modules is also not completely available, as only some of the signaling pathways and protein complexes are experimentally verified. In the following sections, typical evaluation measures are reviewed.

2.9.1 Evaluation of protein function prediction

For individual protein function prediction, a re-sampling method, k-fold cross-validation, is applied to only the known proteins to estimate the performance. The known proteins are randomly divided into k folds or groups. The supervised method learns from $k-1$ folds and makes prediction on the remaining fold, called a "validation set", which is intended as set of unknown proteins. The k-fold cross-validation is applied for k times, and each of the k folds takes turn to be the validation set. The final performance Per is calculated as an average of the performance from k validations $Per_i, i = 1..k$.

$$Per = \frac{1}{k}\sum_{i=1}^{k} Per_i \tag{2.7}$$

		True label	
		g	not g
Prediction label	g	True positive (TP)	False positive (FP)
	not g	False negative (FN)	True negative (TN)

Table 2.3: Contingency table for single functional label prediction

The CAFA challenge of protein function prediction (Radivojac *et al.*, 2013) required a further test of a prediction method on a "test set" of unknown proteins, of which functional labels are unknown at the point of prediction. After a period of time collecting new experimental evidences, some of the test-set proteins become annotated, and the performance on these newly annotated proteins can be computed based on the previous prediction. This can be a future work of our research.

Methodology has long been developed as to evaluating prediction of a single functional label g (Table 2.3). As mentioned in the definition of multiclass prediction, the prediction of functional labels $\hat{L}$ depends on a threshold $0 \leq t \leq 1$ for the posterior probability of a protein p classified to function g. For protein p, the functional labels $\{g_i\}$ of posterior probability above t are chosen as prediction labels.The numbers of FN, TN in the prediction labels will increase if t increases; the numbers of FP, TP will increase if t decreases.

$$\hat{L} = \{g|\hat{\Pr}(g|p) > t\}, p \in U, g \in G \tag{2.8}$$

From the contingency table for functional label g, several performance measures can be calculated. Usually a measure that balances two performance measures is preferred, such as F1 measure or area under curve (AUC) of receiver operating characteristics (ROC).

$$\begin{array}{lcl} \text{precision} & = & \frac{\text{TP}}{\text{TP+FP}} \\ \text{recall} & = & \frac{\text{TP}}{\text{P}} \\ \text{false positive rate (FPR)} & = & \frac{\text{FP}}{\text{N}} \\ \text{true positive rate (TPR)} & = & \frac{\text{TP}}{\text{P}} \\ \text{F1} & = & \frac{2\times\text{precision}\times\text{recall}}{\text{precision+recall}} \end{array} \tag{2.9}$$

The F1 measure balances between precision and recall. We use the measure, maximum F1, over all possible threshold t for functional label g.

$$\max \text{F1} \;=\; \max_{0\leq t\leq 1}\{\text{F1}_t\} \;=\; \max_{0\leq t\leq 1}\{\frac{2\times\text{precision}_t\times\text{recall}_t}{\text{precision}_t+\text{recall}_t}\} \tag{2.10}$$

The ROC curve is a series of $(\text{FPR}_t, \text{TPR}_t)$ points in a two-dimensional space of false positive rate versus true positive rate (Figure). A pair of FPR_t and TPR_t is calculated from one prediction for all unknown proteins. To generate a ROC curve, a set of $T = \{t|0 \leq t \leq 1\}$ is applied. If different thresholds are set for posterior probabilities, multiple sets of predictions can be generated, and multiple (FPR_t, TPR_t) points can be computed to construct the ROC curve. The AUC-ROC measure for functional label g is calculated based on the curve, which balances between FPR and TPR of g.

When evaluating multiclass prediction (i.e. prediction of more than one functional labels, but not necessarily of multi-labels for an instance), the performance measure, such as F1 score or AUC-ROC, are usually calculated per functional label, the same as for single-label binary prediction. Then, the performance is averaged over all $|G|$ labels and further averaged over k folds. Some proposed to calculate an average performance weighted by the prior distribution of the functional labels

$FuncPrior_j, j = 1..|G|$ so that each positive prediction is expected to weigh equally (Fawcett *et al.*, 2006).

$$Per = \frac{1}{k}\sum_{i=1}^{k}(\frac{1}{|G|}\sum_{j=1}^{|G|} FuncPrior_j Per_{ij}) \tag{2.11}$$

In our evaluation of predictions, we disregard the prior weights but expect better evaluation results if prior weights are used in calculating the performance measure. For a performance average, the "one standard error of mean" can be computed over k-fold validation by $s/\sqrt{k}$, where s is the sample standard deviation over k folds.

2.9.2 Evaluation of protein interaction functional modules

For the functional modules of proteins and interactions, because "true" modules are not known, and unknown proteins from the protein modules cannot be validated, the evaluation emphasizes on the proportion of functional modules that are biologically meaningful, and the proportion of functional labels that are predicted.

$$\text{specificity} = \frac{\text{number of protein interaction modules with enriched functional labels}}{\text{total number of protein interaction modules}} \tag{2.12}$$

$$\text{sensitivity} = \frac{\text{number of functional labels enriched in some modules}}{\text{total number of functional labels}} \tag{2.13}$$

3. FINDING CONSERVED FUNCTIONAL MODULES USING GRAPHLET ALIGNMENT OF PROTEIN-PROTEIN INTERACTION NETWORKS (GRAPHLETALIGN)

To investigate the function of genes, one of the most popular strategies is through studying conservation of patterns and modules in biological networks, which can reveal signaling pathways, protein complexes and functional modules. One important strategy is through the use of network alignment techniques, which include heuristic algorithms (Sharan *et al.*, 2005; Koyutrk *et al.*, 2006; Kalaev *et al.*, 2009), progressive alignment algorithms (Flannick *et al.*, 2006), and parameter learning algorithms (Flannick *et al.*, 2009). Although these algorithms are able to identify biologically conserved regions (Kelley *et al.*, 2003; Berg and Lssig 2004), they do not enforce strict topological constraints and it is difficult to use them to study the relationships between topology and function. Recently, the availability of global network alignment algorithms allows these conservation studies to be performed globally across multiple networks (Singh *et al.*, 2008; Flannick *et al.*, 2009; Liao *et al.*, 2009; Kuchaiev and Prulj, 2011).

An alternative strategy to analyze biological networks is through the identification of network motifs (Milo *et al.*, 2002; Wuchty *et al.*, 2003), which are over-represented patterns in a network. While most previous approaches on finding network motifs focus on estimating the number of motifs that have a certain topology in biological networks, either through counting (Shen-Orr *et al.*, 2002; Parida, 2007; Alon *et al.*, 2008), sampling (Kashtan *et al.*, 2004; Jiang *et al.*, 2006; Prulj *et al.*, 2006; Wernicke, 2006), or a combination of these techniques (Grochow and Kellis, 2007), functional linkages of proteins within a motif are ignored.

We investigate the problem of identifying conserved patterns in protein interaction networks by obtaining graphlet alignments, which consist of at least two vertex-disjoint subgraphs that share a common topology and contain homologous proteins at the same position in the topology. Since each topology is represented by a small graphlet, we employ exhaustive enumeration techniques to identify all alignments, which is different from most network alignment algorithms that employ heuristics. By placing a constraint on homology between aligned proteins, our strategy is different from counting the number of network motifs that have a certain topology.

We apply this strategy to protein interaction networks both within and across species, and show that our algorithm is able to cover significantly more proteins in the given networks than previous approaches while maintaining comparable or higher sensitivity and specificity with respect to functional enrichment.

3.1 Methods

We first assume that a single interaction network is given, and our goal is to identify all graphlet alignments within the network (see Figure 3.1 for an illustration).

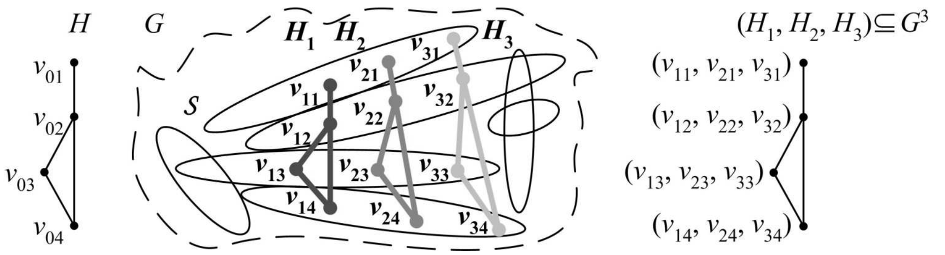


Figure 3.1: Illustration of an interaction network G, a topology H, a graphlet alignment that contains $m = 3$ instances H_1, H_2 and H_3, and its correspondence with an induced subgraph of G^3. Each oval represents one set S_k of homologous proteins in $\mathcal{S}$.

Definition 1. Given an undirected graph G and an integer m, G^m is an undirected graph in which each vertex $(v_1, \ldots, v_m)$ corresponds to an m-tuple of vertices in G, and each edge connects $(v_1, \ldots, v_m)$ and $(v'_1, \ldots, v'_m)$ if (v_i, v'_i) are edges in G for all $1 \leq i \leq m$.

Definition 2. Given an undirected graph $G = (V, E)$ that represents an interaction network, a connected undirected graph $H = (V_0 = (v_{01}, \ldots, v_{0|V_0|}), E_0)$ that represents the target topology, a collection $\mathcal{S} = \{S_1, \ldots, S_{|\mathcal{S}|}\}$ of sets $S_k \subset V$ of homologous proteins, and an integer m, a graphlet alignment consists of m vertex-disjoint instances $H_1, \ldots, H_m$, in which each $H_i = (V_i = (v_{i1}, \ldots, v_{i|V_0|}), E_i)$ is a subgraph of G that is isomorphic to H, and $v_{0j}, v_{1j}, \ldots, v_{mj}$ are mapped together by isomorphism for $1 \leq j \leq |V_0|$, with the restriction that the induced subgraph of $\{(v_{11}, \ldots, v_{m1}), \ldots, (v_{1|V_0|}, \ldots, v_{m|V_0|})\}$ on G^m is isomorphic to H, and for each j, $v_{1j}, \ldots, v_{mj}$ are homologous proteins, that is, $\{v_{1j}, \ldots, v_{mj}\} \subseteq S_k$ for some k.

Similar to the motif discovery approach in Grochow and Kellis (2007), we assume that a fixed topology is given, and our algorithm can be applied over different topologies to identify all graphlet alignments. Since the given topology does not contain actual proteins, this strategy is different from other approaches that identify conserved patterns through a query path or a query network (Kelley *et al.*, 2003; Shlomi *et al.*, 2006; Tian *et al.*, 2007; Dost *et al.*, 2008). Since no restrictions are made on the relationships between different sets S_k of homologous proteins, they can overlap in complicated ways. This strategy is different from the one in Koyutrk *et al.* (2004), in which the same label is used to specify homologous proteins in different graphs. In addition to subgraphs in G, the isomorphism mappings are applied to induced subgraphs in G^m to make sure that H is the densest topology that all instances share (note that the subgraphs in G do not need to be induced). To avoid repetitive structures, we require that each vertex in G can only appear at most once within an alignment. Note that this does not prevent different homologous proteins from appearing within an instance.

We exhaustively find all graphlet alignments through a branch-and-bound algorithm by generating m-tuples of vertices in G that satisfy the homology constraints and recursively adding them to a growing alignment with m instances. This technique updates the m instances at the same time and is different from the progressive alignment technique used in network alignment algorithms (Flannick *et al.*, 2006). To avoid the generation of redundant alignments, we impose symmetry breaking conditions.

Grochow and Kellis (2007) considered the problem of enumerating all subgraphs of an undirected graph G that are isomorphic to a given topology H, and derived symmetry breaking conditions to ensure that there is a unique map from H to each instance of H in G. Given a distinct labeling of vertices in G, they imposed conditions of the form $l_H(v) < \min(l_H(u_1), \ldots, l_H(u_k))$, where $v, u_1, \ldots, u_k$ are vertices of H and $l_H(v)$ is the induced label of vertex v in H given a map from H to G.

To impose symmetry breaking conditions on graphlet alignments, observe that there is a one-to-one correspondence between a graphlet alignment with m instances and an induced subgraph in G^m that contains m-tuples of homologous proteins as vertices with no repeated vertices in G. For each j, v_{ij} of H_i in a graphlet alignment corresponds to the ith component of the vertex $(v_{1j}, \ldots, v_{mj})$ in G^m that represents a set of m aligned proteins. Note that v_{ij} of H_i is mapped from the jth vertex of H, and a map from H to G^m specifies an alignment with m instances (see Figure 3.2). Note, for example, that (u, v) and (v, u) are two distinct vertices in G^2.

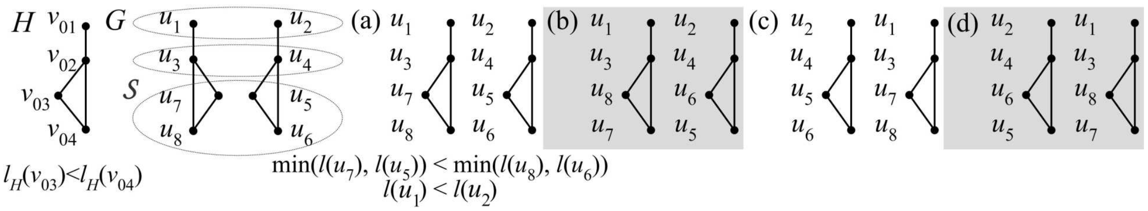

H
v_{01}
v_{02}
v_{03}
v_{04}
G
u_1
u_2
u_3
u_4
S
u_7
u_5
u_8
u_6
(a) u_1 u_2 u_3 u_4 u_7 u_5 u_8 u_6
(b) u_1 u_2 u_3 u_4 u_8 u_6 u_7 u_5
(c) u_2 u_1 u_4 u_3 u_5 u_7 u_6 u_8
(d) u_2 u_1 u_4 u_3 u_6 u_8 u_5 u_7
$l_H(v_{03})<l_H(v_{04})$
$\min(l(u_7), l(u_5)) < \min(l(u_8), l(u_6))$
$l(u_1) < l(u_2)$

Figure 3.2: Illustration of redundant alignments. The alignments in (a) and (b), and the alignments in (c) and (d) are symmetric in topology, while the alignments in (a) and (c), and the alignments in (b) and (d) have their instances permuted. The symmetry breaking condition on H is shown below H. The corresponding symmetry breaking conditions on (a) are shown below (a), in which the first condition is obtained by setting $l_H(v_{03}) = l(u_7, u_5) = \min(l(u_7), l(u_5))$ and $l_H(v_{04}) = l(u_8, u_6) = \min(l(u_8), l(u_6))$ to break symmetry of topology, and the second condition is obtained from the first 2-tuple (u_1, u_2) to break symmetry due to permutation of instances.

The problem reduces to enumerating all induced subgraphs of G^m that are isomorphic to H and contain m-tuples of homologous proteins as vertices with no repeated vertices in G, thus the symmetry breaking conditions in Grochow and Kellis (2007) can be used to break symmetry of topology in G^m. We have an additional type of symmetry due to permutation of instances.

Definition 3. Two graphlet alignments with m instances are redundant if they have exactly the same sequence of m-tuples of homologous proteins up to symmetry of topology in G^m and symmetry due to permutation of instances (see Figure 3.3).

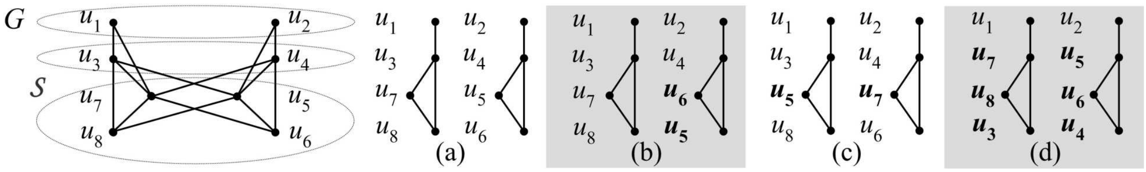


Figure 3.3: Illustration of redundant alignments. The alignments in (a) and (b), and the alignments in (c) and (d) are symmetric in topology, while the alignments in (a) and (c), and the alignments in (b) and (d) have their instances permuted. The symmetry breaking condition on H is shown below H. The corresponding symmetry breaking conditions on (a) are shown below (a), in which the first condition is obtained by setting $l_H(v_{03}) = l(u_7, u_5) = \min(l(u_7), l(u_5))$ and $l_H(v_{04}) = l(u_8, u_6) = \min(l(u_8), l(u_6))$ to break symmetry of topology, and the second condition is obtained from the first 2-tuple (u_1, u_2) to break symmetry due to permutation of instances.

To break both types of symmetry, we first make sure that all alignments that differ only by a permutation of instances are treated in exactly the same way with respect to topology symmetry breaking by assigning the same label to each m-tuple $(v_1, \ldots, v_m)$ in G^m and all its permutations $(v_{p(1)}, \ldots, v_{p(m)})$, where p is an arbitrary permutation on m vertices. One way to do this is to assign the label $l(v_1, \ldots, v_m) = \min(l(v_1), \ldots, l(v_m))$ to each m-tuple $(v_1, \ldots, v_m)$ in G^m, where $l(v)$ is the distinct label of vertex v in G (see Figure 3.2). Note that this does not mean that these m-tuples are treated in the same way. They are distinct in G^m and can appear in different alignments (see Figure 3.3). Also note that since no repeated vertices in G are allowed in an alignment, topology symmetry breaking is performed with respect to m-tuples that have distinct labels. Redundant alignments that differ only by a permutation of instances are removed by imposing the condition $l(v_{11}) < \cdots < l(v_{m1})$ on the first m-tuple of the instances $H_1, \ldots, H_m$ (see Figure 3.2).

Figure 3.4 shows our algorithm GraphletAlign that enumerates all non-redundant graphlet alignments by applying the procedure ExtendAlignment to grow an alignment recursively (Figure 3.5). Unlike other network alignment approaches that combine more than one network into a single graph during preprocessing (Kelley *et al.*, 2003; Sharan *et al.*, 2005; Koyutrk *et al.*, 2006), there is no need to generate the entire graph G^m explicitly, which is not feasible when m is large due to extensive space requirements. During the generation of m-tuples $(v_1, \ldots, v_m)$, it is necessary to consider only vertex combinations within each S_k in $\mathcal{S}$ that are of size at least m. We order vertices in G and H in such a way to reduce unsuccessful branches that will need to be pruned later.

Figure 3.4: Algorithm GraphletAlign for finding graphlet alignments with m vertex-disjoint instances $H_1, \ldots, H_m$ when given an interaction network G, a topology H, and a collection $\mathcal{S}$ of sets of homologous proteins.

```
**Algorithm** GraphletAlign($G, H, \mathcal{S}, m$)
**begin**
    Assign label $l(v)$ to each vertex $v$ of $G$;
    Determine topology symmetry breaking conditions of $H$;
    **for each** vertex-disjoint $m$-tuple $(v_1, \ldots, v_m)$ of $G^m$
      with label $\min(l(v_1), \ldots, l(v_m))$ that satisfies that:

      (i) There exists $\{v_1, \ldots, v_m\} \subseteq S$, for some $S \in \mathcal{S}$, and
      (ii) $l(v_1) < \cdots < l(v_m)$

    **do**
        Set the first vertex of $H_i$ to $v_i$ for $1 \leq i \leq m$;
        Call ExtendAlignment($G, H, \mathcal{S}, m, 1$);
    **done**
**end**
```

```
Algorithm ExtendAlignment($G, H, \mathcal{S}, m, j$)
begin
    if $j$ = total number of vertices in $H$
    then
        Store the new alignment with instances $H_1, \ldots, H_m$;
    else
        for each vertex-disjoint $m$-tuple $(v_1, \ldots, v_m)$ of $G^m$
        with label $\min(l(v_1), \ldots, l(v_m))$ that satisfies that:

          (i) There exists $\{v_1, \ldots, v_m\} \subseteq S$, for some $S \in \mathcal{S}$, and
          $\{v_1, \ldots, v_m\}$ is vertex-disjoint from the first $j$ vertices in each of $H_1, \ldots, H_m$;

          (ii) There is an edge between $v_i$ and the $k$th vertex of $H_i$ in $G$,
          for all $1 \leq i \leq m$, if and only if there is an edge between
          the $(j+1)$th vertex and the $k$th vertex of $H$, for $k \leq j$; and

          (iii) The topology symmetry breaking conditions involving
          the $(j+1)$th vertex and previous vertices of $H$ are met,
          given that $(v_1, \ldots, v_m)$ is mapped to the $(j+1)$th vertex of $H$

        do
            Set the $(j+1)$th vertex of $H_i$ to $v_i$ for $1 \leq i \leq m$;
            Call ExtendAlignment($G, H, \mathcal{S}, m, j+1$);
        done
    fi
end
```

Figure 3.5: Algorithm ExtendAlignment for extending alignments that contain the first j vertices of each of the m vertex-disjoint instances $H_1, \ldots, H_m$ to contain one more vertex when given an interaction network G, a topology H, and a collection $\mathcal{S}$ of sets of homologous proteins.

Since for each m-tuple in G^m, the algorithm ExtendAlignment is called recursively on all its adjacent m-tuples at most $|V_0| - 1$ times, the worst case time complexity is $O(|V|^m \Delta^{|V_0|-1})$, where Δ is the largest vertex degree of G^m. This can be rewritten as $O(|V|^m \delta^{m(|V_0|-1)})$, where δ is the largest vertex degree of G. In reality, the number of needed vertices in G^m is bounded by the number of distinct m-tuples of vertices in G that can be obtained from $\mathcal{S}$, and the requirement of finding induced graphs helps to prune a lot of search branches. As long as most vertices in G^m are not close to the maximum degree, the actual computational time should be much lower than the worst case estimate.

To allow indels between neighboring proteins within the instances, given a parameter d that specifies the maximum length of gaps, we construct a new graph $G' = (V', E')$ from $G = (V, E)$ by setting $V' = V$ and connecting vertex u to v in G' if the shortest path distance between u and v is at most $d + 1$ in G, and apply the algorithm to G' instead of G. To allow the identification of graphlet alignments in multiple networks, we combine the given networks into a single graph G (no new edges are added), and assign consecutive labels to vertices in G within each of the networks. We further require that each alignment contains at least one instance from each network.

3.2 Results

We applied the GraphletAlign algorithm on non-isomorphic topologies H (Figure 3.6) obtained from using the algorithm in McKay (1998) to protein interaction networks from human and mouse in the IntAct database (Hermjakob *et al.*, 2004), to protein interaction networks from fly, worm and yeast in the DIP database (Xenarios *et al.*, 2000), and to protein interaction networks from *E. coli*, *H. pylori*, *S. typhimurium* and *V. cholerae* in the SNDB database (Srinivasan *et al.*, 2006). To obtain the collection $\mathcal{S}$ of sets of homologous proteins, for each protein we identify top r BLAST hits (Altschul *et al.*, 1990) with e-value below 10^{-7} so that the protein itself is also a reciprocal BLAST hit to form one set of homologous proteins, where r is a parameter.

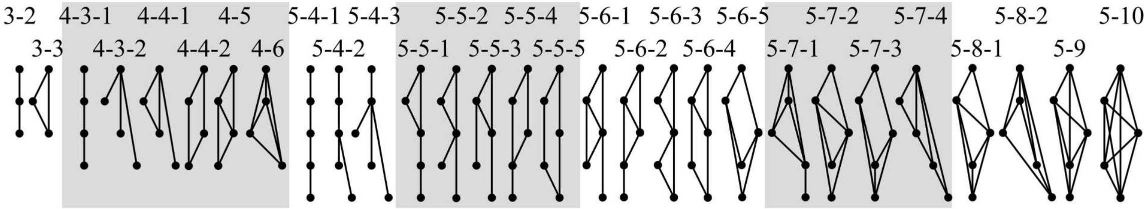


Figure 3.6: Illustration of all non-isomorphic topologies H with three to five vertices. Each topology is assigned a name of the form x-y or x-y-z, where x is the number of vertices in the topology, y is the number of edges, and z distinguishes between topologies that have the same values of x and y.

In order to allow an indel in a graphlet alignment between two sets of m aligned proteins, one from each instance, we impose an additional condition that there is at least one direct connection within one of the instances in the original network. Note that indels are allowed within each edge of the topology according to the way that G' is constructed to replace G. Table 3.1 shows the number of graphlet alignments over a few combinations of species when one indel is allowed. For single species alignments, we use topology size $|V_0| = 3$ and number of instances $m = 2$. For two and three species alignments, we also use topology size $|V_0| = 4$. For two species alignment on fly and yeast, we further use topology size $|V_0| = 5$. For multiple species alignments, we set m to be the number of species. We set the maximum number of top reciprocal BLAST hits r allowed for each protein to the largest possible value (in multiples of five) so that the number of alignments is between 10^7 and 10^8. It takes between half-a-day and two days to obtain all alignments on a single processor over all topologies in each case.

Table 3.1: Number of graphlet alignments and computational time on single species, two species, three species and four species alignments when one indel is allowed.

	Vertices	Edges	$\lvert V_0 \rvert$	m	r	Alignments	Avg. size	Max. size	Time
human	6294	13455	3	2	5	3.0×10^7	45.0	177	883.2
yeast	7446	22734	3	2	10	3.6×10^7	59.0	117	662.2
human–mouse	7455	14545	3–4	2	—	2.2×10^7	21.4	109	1246.8
fly–yeast	12374	40111	3–5	2	15	4.8×10^7	35.1	100	2585.7
fly–worm–yeast	14254	42361	3–4	3	40	3.5×10^7	17.9	100	1011.9
E.col–H.pyl–S.typ–V.cho	10339	64890	3	4	35	7.6×10^7	9.3	41	1635.1

Vertices: The total number of vertices in the given interaction networks

Edges: The total number of edges

$\lvert V_0 \rvert$: The range of size of topology

m: The number of instances

r : The maximum number of top reciprocal BLAST hits allowed for each protein
(with — indicating no constraint)

Alignments: The total number of alignments over all topologies

Avg. size: The average number of distinct proteins within a module instance
after postprocessing (Only aligned proteins are counted while indels are ignored)

Max. size: The maximum number of distinct proteins within a module instance,
and time is the computational time to obtain all alignments in minutes

We compare the performance of our algorithm on multiple species alignments to NetworkBLAST-M (Kalaev *et al.*, 2009), which identifies conserved functionally enriched modules based on a representation of multiple networks that is of linear size. On two species alignments, we further compare to NetworkBLAST (Sharan *et al.*, 2005), which combines more than one given network into a single graph and uses a seed-extension approach to find high scoring subgraphs that represent alignments, and MaWISh (Koyutrk *et al.*, 2006), which considers an evolutionary model that includes interaction matches, interaction mismatches and protein duplication to define high scoring pairwise alignments. On two species alignment of fly and yeast, we also compare to DOMAIN (Guo and Hartemink, 2009), which incorporates information from domain interactions to produce pairwise alignments based on alignment of edges rather than nodes. On three species alignment, we also compare to CAPPI (Dutkowski and Tiuryn, 2007), which identifies functional modules by reconstructing an ancestral network based on a network evolution model. On four species alignment, we also compare to Grmlin (Flannick *et al.*, 2006), which employs a progressive alignment approach that allows a large number of networks to be aligned together. Since our algorithm is the only one among these algorithms that can identify alignments within a single species, no comparisons are performed in this case.

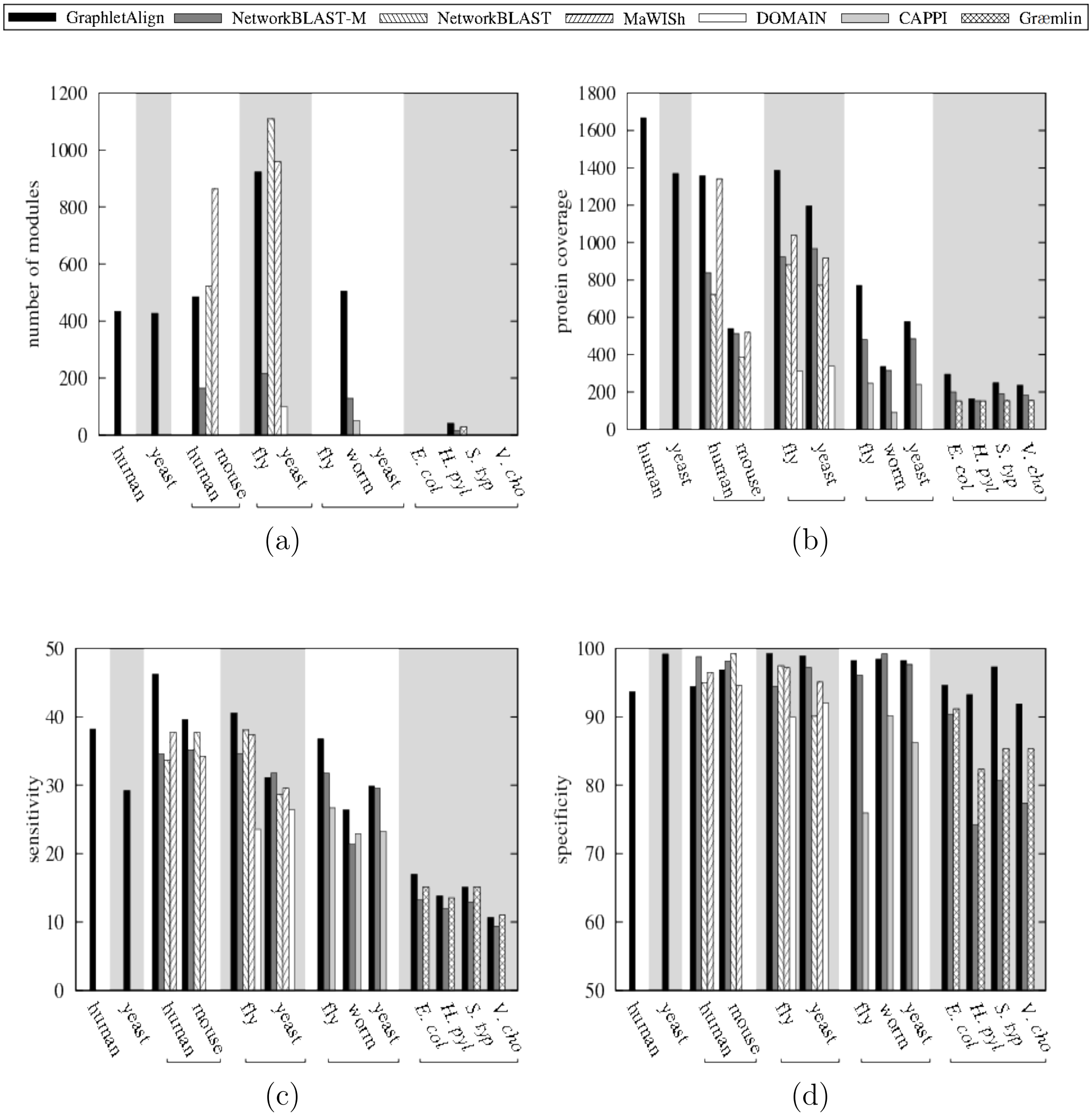
We perform postprocessing to obtain larger modules from the graphlet alignments. In order to avoid exhaustive comparisons between all pairs of alignments when the number of alignments is large, we consider the alignments within each topology in search traversal order and merge all neighboring alignments that have exactly the same m-tuples in G^m at all search levels except for the last level. We merge each instance of an alignment with the corresponding instance of another alignment separately to obtain a module with larger instances.

We perform further postprocessing to reduce the number of modules and overlap between modules while increasing the module size. In order not to lose information from our exhaustive set of graphlet alignments, we merge some of these modules instead of removing them. We define the score of a graphlet alignment to be the average minus log e-value from BLAST of all aligned protein pairs, and the score of a module to be the best graphlet alignment score within the module. For each aligned protein, we construct a list of all the modules that contain the protein. We sort the module lists in decreasing order of the number of modules in each list and consider each module list in order. Within each module list, we sort the modules in decreasing order of the number of aligned proteins in the module, then in decreasing order of the number of aligned edges within the ordering, and finally in decreasing order of the module score. We iteratively consider the highest ranked module in the list. We compare it to the next lower ranked module and merge them if the merged module does not contain more than 100 aligned proteins within an instance and the overlap of aligned proteins between each pair of corresponding instances is at least 50% with respect to the average number of aligned proteins in the instance pair. We remove the lower ranked module after merging and replace the highest ranked module with the merged module. We continue to compare the highest ranked module with the next lower ranked module until there are no more lower ranked modules to compare, at which time we remove the module lists of all the aligned proteins that are contained in the highest ranked module. We continue with the next module list until there are no more module lists left. Table 3.1 shows that the average size of the modules remains small after merging (although the maximum module size can be large).

To investigate functional relationships among aligned proteins within a module, we consider each species separately, and use gene ontology (GO) annotations (Ashburner *et al.*, 2000) to determine whether its aligned proteins tend to have related function. We evaluate its functional enrichment by applying the GO Term Finder (Boyle *et al.*, 2004) to the aligned proteins and identifying significant GO terms with Bonferroni corrected p-value below 0.05 within the biological process ontology. We define the specificity to be the percentage of modules that have significant GO terms within each species. We map each significant GO term to all ancestral GO terms with a shortest path distance of two from the root of the biological process ontology, which represent a subset of high-level GO terms that represent functional categories. We define the sensitivity to be the percentage of these ancestral GO terms that are mapped from at least one significant GO term.

Figure 3.7 shows that our algorithm was able to cover significantly more proteins.

Except for a few cases, our algorithm usually had higher sensitivity and specificity than the other algorithms. Similar to NetworkBLAST and MaWISh, our algorithm returned a larger number of modules than the other algorithms. This is due to the condition that two modules have to satisfy the overlap threshold within each species in order to be merged, so that modules that occupy very different regions within some species will remain separate. Both the protein coverage and sensitivity decrease as the number of species increases. In general, the number of modules is highly correlated to the number of alignments. It is necessary to allow indels since otherwise protein coverage is much lower. While it is not feasible to use all BLAST hits, protein coverage can be improved by increasing the maximum number of top reciprocal BLAST hits allowed for each protein. When the number of instances is small, it is necessary to consider larger topologies to maintain high sensitivity and specificity.



(a)

(b)

(c)

(d)

Figure 3.7: Performance comparisons of GraphletAlign, NetworkBLAST-M, NetworkBLAST, MaWISh, DOMAIN, CAPPI and Græmlin on single species, two species, three species and four species alignments. For GraphletAlign, one indel is allowed and parameter settings are in Table 3.1. For the other algorithms, the same networks and the same BLAST e-value threshold are used. (a) Number of modules. (b) Protein coverage: the total number of distinct proteins that are covered by these modules within each species (only aligned proteins are counted while indels are ignored). (c) Sensitivity: the percentage of functional categories as defined by all ancestral GO terms with a shortest path distance of two from the root of the biological process ontology that are mapped from at least one significant GO term within each species. (d) Specificity: the percentage of modules that have significant GO terms within each species while excluding the ones that do not have any GO term annotations.

To investigate whether completely different conserved regions can be obtained from different algorithms within each species, we consider each algorithm and retain only the modules in which all proteins within at least one species are not covered by another algorithm. Note that these modules can still overlap with proteins covered by the other algorithm within some other species. Figure 3.8 shows that our algorithm was able to identify some number of such modules with respect to each of the other algorithms and the specificity of these modules remains high, while the other algorithms generally identified fewer such modules with respect to our algorithm.

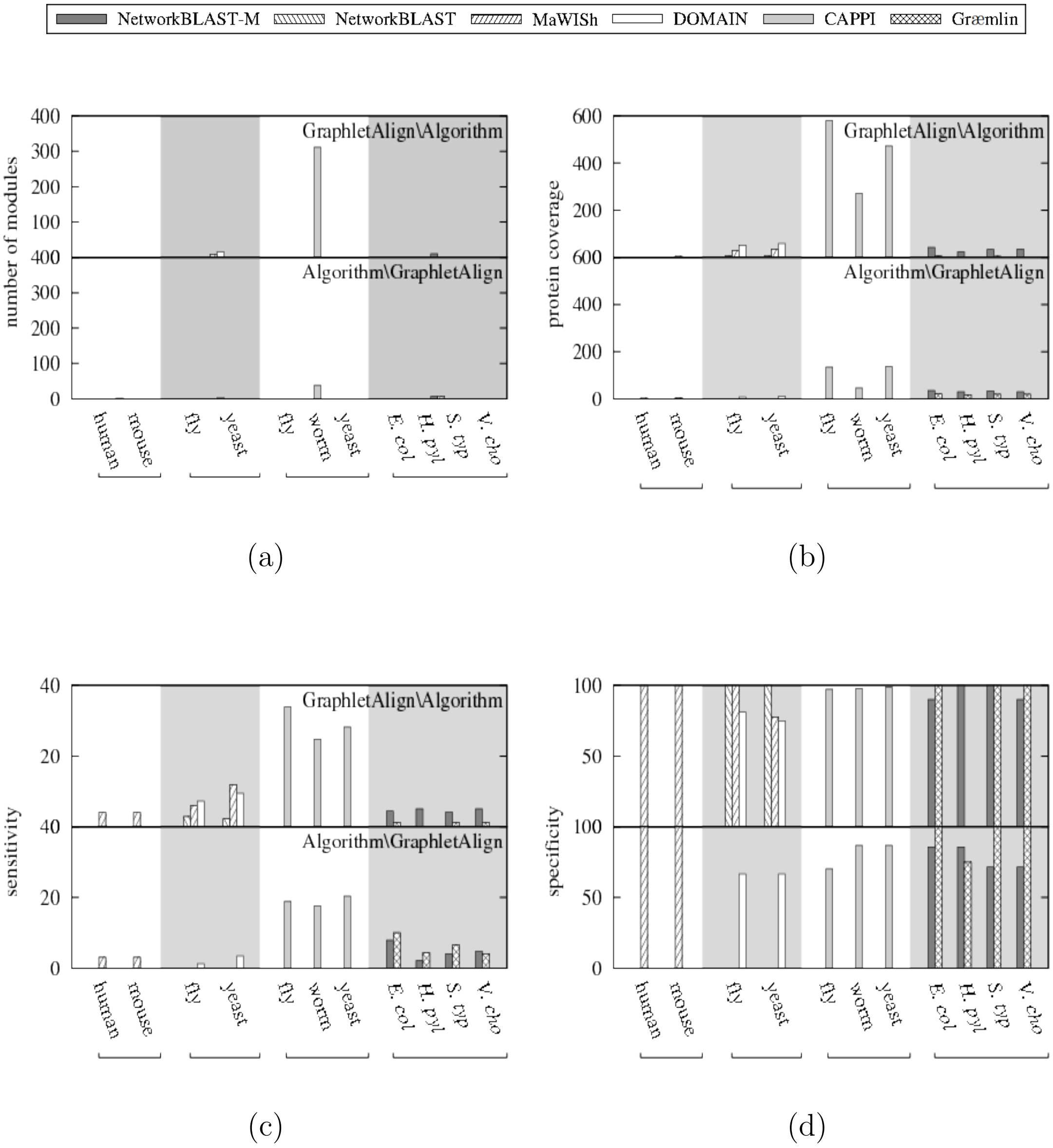


Figure 3.8: Performance comparisons between GraphletAlign and each of the algorithms NetworkBLAST-M, NetworkBLAST, MaWISh, DOMAIN, CAPPI and Grmlin on multiple species alignments when retaining only the modules in which all proteins within at least one species are not covered by another algorithm. The notation X\Y denotes the performance of algorithm X with respect to another algorithm Y. Each graph shows the same statistics as in Figure 3.7 except that they are only on the retained modules. The notation in (d) is the same as the other graphs.

Figure 3.9 shows four graphlet alignments found by our algorithm in the yeast network that link together three mitogen-activated protein (MAP) kinase cascades in the pheromone response pathway, the filamentation/invasion pathway, and the cell integrity pathway, in which the correspondences between the MAPKK, MAPK and MAP kinases are all in the correct positions (Gustin *et al.*, 1998). Figure 3.10 shows a module found by our algorithm but not by NetworkBLAST-M that contains cold shock proteins nusA, infB, pnp and rpsO (Bae *et al.*, 2000), with a strong relationship of the extra protein fusA in *H. pylori* to cold shock response (Delgado *et al.*, 2008).

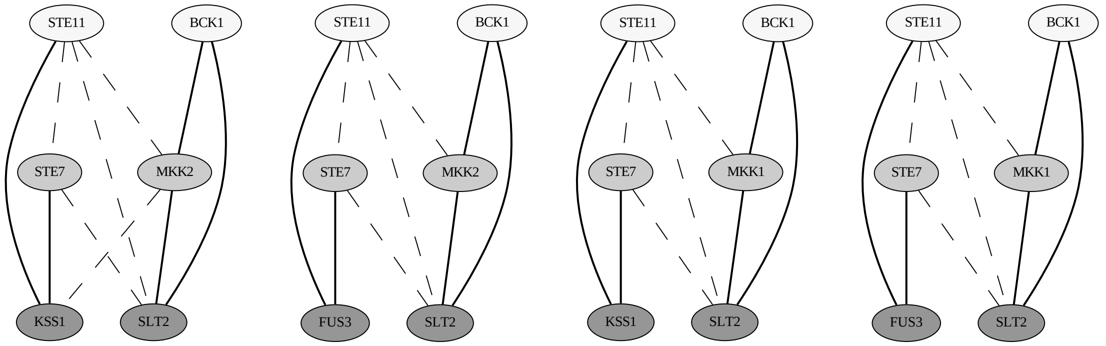


Figure 3.9: Mitogen-activated protein (MAP) kinase cascades found in the graphlet alignments of yeast with topology 3-3. Solid lines denote direct interactions, while dashed lines denote indirect interactions. Proteins within the same row are aligned together.

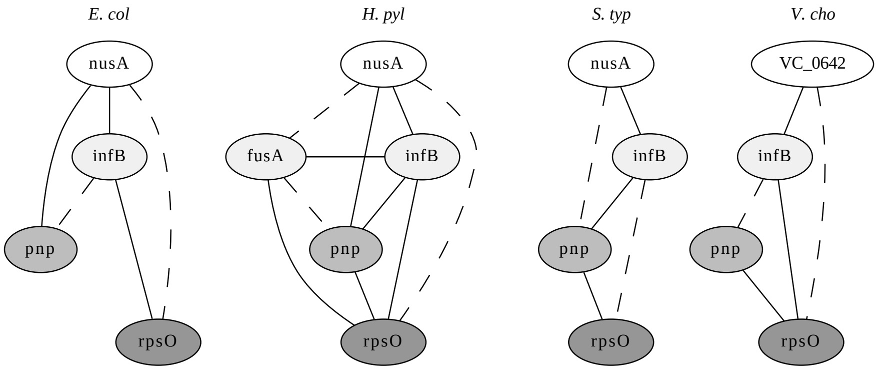
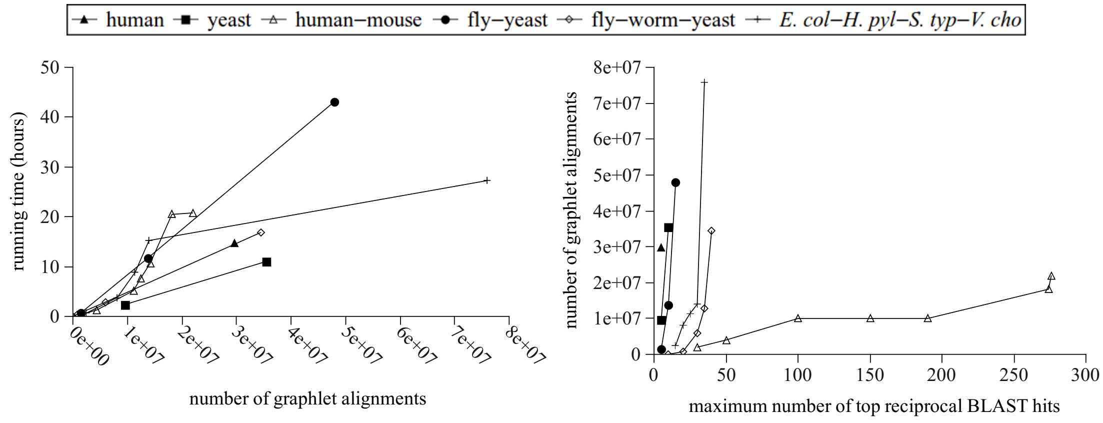


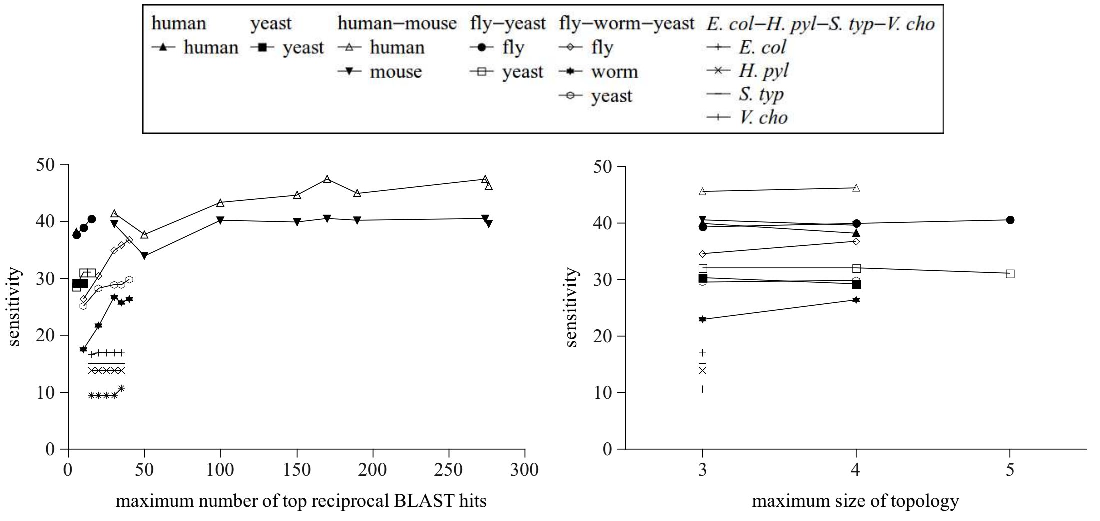
Figure 3.10: A module found by GraphletAlign but not by NetworkBLAST-M that contains cold shock proteins. Solid lines denote direct interactions, while dashed lines denote indirect interactions.

Although our algorithm is slower than previous algorithms, its running time is approximately linear in the number of graphlet alignments that are generated, while the number of graphlet alignments is highly correlated to the maximum number of top reciprocal BLAST hits r allowed for each protein (see Figure 3.11). Figures 3.12 and 3.13 further show that both sensitivity and specificity increase as r increases, but they level off after r becomes large enough. As the maximum size of topology $|V_0|$ that is used increases, sensitivity stays relatively constant while specificity gradually increases.



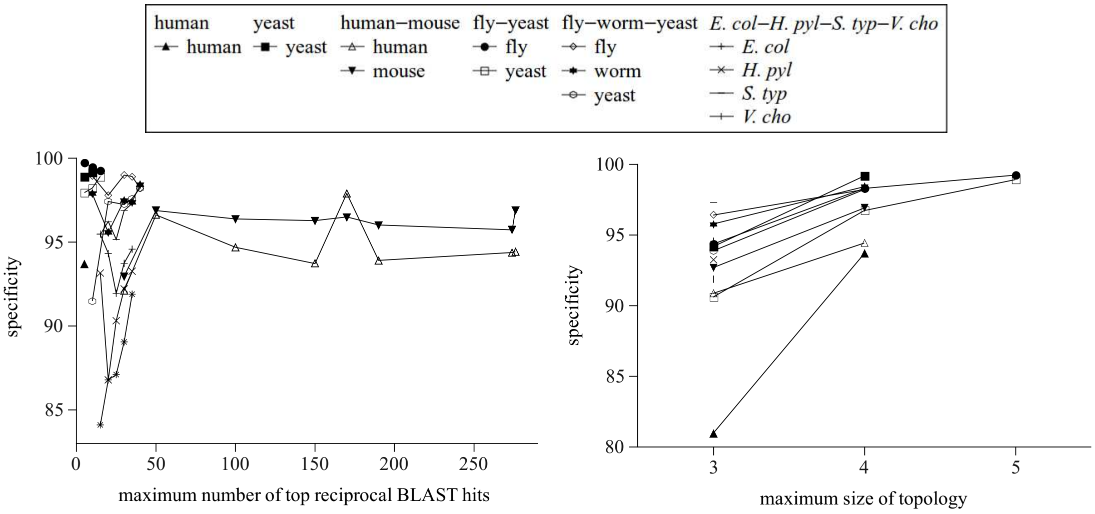
(a) (b)

Figure 3.11: Computational statistics of GraphletAlign. (a) Running time as a function of the number of graphlet alignments that are generated. (b) Number of graphlet alignments as a function of the maximum number of top reciprocal BLAST hits r allowed for each protein.



(a) (b)

Figure 3.12: Sensitivity of GraphletAlign against (a) the maximum number of top reciprocal BLAST hits r allowed for each protein and (b) the maximum size of topology $|V_0|$ that is used.



(a) (b)

Figure 3.13: Specificity of GraphletAlign against (a) the maximum number of top reciprocal BLAST hits r allowed for each protein and (b) the maximum size of topology $|V_0|$ that is used.

4. PROTEIN FUNCTION PREDICTION FROM GENETIC AND PROTEIN-PROTEIN INTERACTIONS VIA REGULARIZED MULTI-LABEL CANONICAL DISCRIMINANT ANALYSIS (MCDA)

Proteins are often annotated with one or more functions (i.e., multi-labeled), as seen in gene ontology (GO) (Ashburner *et al.*, 2000), enzyme classifications (Kanehisa *et al.*, 2004) and the MIPS Functional Catalogue (FunCat) (Mewes *et al.*, 2004). Many previous algorithms have been developed that utilize interaction network data to predict protein function (Vazquez *et al.*, 2003; Karaoz *et al.*, 2004; Nabieva *et al.*, 2005), while integrated approaches have also been developed to utilize more than one source of data (Lanckriet *et al.*, 2004; Tsuda and Noble, 2004; Mostafavi *et al.*, 2008). Multi-label classification methods consider association between groups and have been shown to outperform classical single-label binary classifier (Wang *et al.*, 2015), which assumes that each group is independent and classifies one single group at a time. We introduce multi-label canonical discriminant analysis (MCDA), which is a generalization of the classical multiclass classifier, canonical discriminant analysis (CDA).

Discriminant analysis searches for the linear combinations of explanatory variables (discriminant functions) that best discriminate between the groups defined (Legendre & Legendre, 2012). Further, relative contributions of the explanatory variables to the discriminant functions can be determined. The resulting functions can be used to predict group membership for sampling entities of unknown membership. Discriminant analysis is used by researchers in a wide variety of fields including biological and earth sciences, business, economics, sociology, psychology, medical diagnostics, public health, education, engineering, political science and other disciplines.

In biology, discriminant analysis is used in many kinds of research (McGarigal *et al.,* 2000). In biological systematics, it has long been considered desirable to be able to discriminate among groups of organisms (populations, species, etc.) on the basis of morphological characters and molecular datasets (Moder *et al.,* 2007, Kim *et al.,* 2010, Jaiswara *et al.,* 2013, Benca *et al.,* 2014, Reif *et al.,* 2015). Discriminant analysis was also performed to determine which characters contributed most to the separation of the taxa. However, to assign new objects to previously separated groups was used less frequently in systematics (Kouteck 2007, Jaiswara *et al.,* 2013).

Discriminant analysis has been applied extensively to ecological datasets, including vegetation science (Salovaara *et al.,* 2005, Thessler 2008, Kusbach *et al.,* 2015), landscape ecology (L´opez-Granados *et al.,* 2010, Petr and Mikita 2011; Abdallah *et al.,* 2015, Phompila *et al.,* 2015), climate change (Braun *et al.,* 2013) and palaeoecology (Nfrdi *et al.,* 2014) for many decades. It has been proven to be a powerful and reliable method to produce reasonably accurate habitat and vegetation classifications based on species distribution pattern or remote sensing data, and the resultant functions can then be used to predict class membership for unvisited sampling units or the entire study area (Legendre & Legendre, 2012). This approach has even been applied in structurally relatively homogeneous lowland tropical forests (Tuomisto *et al.*, 2003). Besides classification, discriminant analysis can also be used to assessed the relationships between classification units and physical environment (McGarigal 2000, Kusbach *et al.*, 2015).

Among the methods of discriminant analysis, canonical discriminant analysis (CDA), a variant of linear discriminant analysis (LDA), has been used by ecologists to classify species based on numerical features, such as soil properties, weather conditions, terrain types, and also classify based on similarity or distance matrices, such as geographical correlations between species, when combined with multi-dimensional scaling (MDS). Despite its popularity in these areas, CDA has not yet been applied to protein function prediction.

The classic CDA takes input: a set of m multivariate Gaussian population groups $g_1, g_2, \ldots, g_m$ of variables x with means $\mu_1, \mu_2, \ldots, \mu_m$ and a common covariance matrix Σ, and CDA predicts the probability of x belonging to each group g_j, where each variable x can belong to only one group g_j. Discriminant functions are derived through eigen-decomposition of the product of the estimated between-class and within-class covariance matrices to maximize the separation of population groups at transformed space, where a p-dimensional variable x is transformed to a q-dimensional variable z for $q \leq p$. The probability of a variable z belonging to a group g_j is predicted based on its distance to group centroids at the transformed space (also called canonical space for CDA). CDA differs from LDA majorly on how the covariance matrices are estimated, with CDA relying on multivariate analysis of variance (MANOVA).

For a traditional multiclass classifier, such as CDA, each variable may belong to one of the multiple classes (i.e. groups). In multi-label classification, each variable may belong to one or more groups among $g_1, g_2, \ldots, g_m$. In our multi-label CDA, given n variables $x_1, x_2, \ldots, x_n$, the group membership is represented by a design matrix D, where $D_{ij} = 1$ indicates that variable x_i belongs to group g_j. While a previous multi-label LDA approach (Wang *et al.*, 2010) computes a between-class covariance matrix and a pooled within-class covariance matrix by summing up the corresponding covariance matrices of each class/group in both cases, our multi-label CDA computes a single within-class covariance matrix by MANOVA as in classic CDA.

One difficulty of the approach is that when the number of classification features is larger than the cardinality of any group, collinearity problem may occur and result in a singular covariance matrix. Furthermore, the functional labels in the design matrix may be correlated, and the regression residuals calculated from MANOVA may be correlated, and all these may create the singularity problem for covariance matrix. Besides replacing the matrix inverse with pseudo-inverse, a commonly used regularization technique can be applied to stabilize the covariance matrix and reduce the bias of discriminant functions before matrix inversion (Guo *et al.*, 2007). We propose a two-step regularization for MCDA, which includes applying ridge regression (Hoerl and Kennard, 1970) to regularize the between-class covariance matrix, and regularizing the within-class covariance matrix directly.

The other challenge is that it is hard to quantify the protein features such as sequences and interactions directly into predictor of protein functions. In contrast, there have been many researches that compare sequences or neighborhood interactions. We propose to follow an analysis framework used by ecologists: defining distance matrix based on observations, applying multi-dimensional scaling (MDS) to transform the distance matrix to feature vectors, and applying CDA to transform the MDS variables to canonical variables for classification.

We apply the framework along with regularized MCDA to the multi-label protein function prediction problem. We compare our performances to a multi-label classifier, the maximization of data-knowledge consistency (MDKC) approach, which formulates the protein function prediction as a non-negativity matrix factorization problem to minimize the difference between data similarity and functional label similarity (Wang *et al.*, 2015). MDKC also considers functional group correlation in the problem formulation. We also compare to the multiclass binary classifiers tested by the MDKC paper, including the multiple-kernel and SVM approach (Lanckriet *et al.*, 2004), multiple-kernel and min-max approach (Tsuda and Noble, 2005), the global neighborhood optimization and minimum multiway cut approach (Vazquez *et al.*, 2003), and the local neighborhood counting method (majority voting) (Schwikowski *et al.*, 2000).

4.1 Methods

4.1.1 Multi-label protein function prediction

We define the multi-label protein function prediction problem as follows. Given n sample proteins $S = \{p_1, p_2, \ldots, p_n\}$, observed biological features of these proteins F, m protein functional groups $G = \{g_1, g_2, \ldots, g_m\}$, and a pair set that associates proteins and group labels $L = \{\ldots, \{p_i, g_j\}, \ldots\}$ for known proteins $p_i \in K \subset S$, predict the pair set $L_U = \{\ldots, \{p_i, g_j\}, \ldots\}$ for each of the unknown protein $p_i \in U = (S - K)$ that maximizes the posterior probability distribution function (p.d.f.) over all possible functional groups (Radivojac *et al.*, 2013).

$$\hat{L_U}(p) = \arg\max_{g \in G}\{\hat{\Pr}(g|p)\}, p \in U \tag{4.1}$$

Here, the observations F may include protein sequences, genetic interaction networks, protein-protein interaction networks or other type of data.

A prediction method may output the posterior probabilities $\hat{\Pr}(g|p)$ that each unknown protein p is assigned to a functional group g. By setting a threshold $0 \leq t \leq 1$ for the posterior probabilities, the labels of each protein p can be determined.

$$\hat{L_U}(p) = \{g|\hat{\Pr}(g|p) > t\}, p \in U \tag{4.2}$$

4.1.2 Distance matrices

While various pairwise distance measures have been developed for genes, proteins or species in phylogeny, a distance matrix can also be calculated based on the topologies in a genetic or protein-protein interaction network. For example, the graph adjacency matrix can be viewed as a basic similarity matrix for genes/proteins. A measure of topological similarity was also developed for every two genes/proteins in the network (Pei and Zhang, 2005).

In our protein function prediction algorithm, we define distance matrices from sequence and interaction network comparisons. The sequence distance between two sequences s_i and s_j is based on the PAM similarity in a Smith-Waterman local alignment (Smith and Waterman, 1981).

$$\mathrm{PAM}(s_i, s_j) = 10 \times (10\hat{\ } Pr(\frac{s_i \text{ and } s_j \text{ have common ancestor}}{s_i \text{ and } s_j \text{ are random alignment}}) - 1) \tag{4.3}$$

We adopt the following pairwise distance SW of proteins p_i and p_j, which is calculated from the PAM similarity of their corresponding sequences s_i and s_j.

$$\begin{aligned} \mathrm{SW}(p_i, p_j) &= 1 - Pr(\tfrac{s_i \text{ and } s_j \text{ have common ancestor}}{s_i \text{ and } s_j \text{ are random alignment}}) \\ &= 1 - \log_{10}(\mathrm{PAM}(s_i, s_j)/10 + 1) \end{aligned} \tag{4.4}$$

The interaction distance between two proteins is defined as one minus the similarity of neighborhood interactions. We adopt the Srensen-Dice (SD) similarity or Jaccard (Jc) similarity to represent common interaction partners of the two proteins, following the intuition in Bandyopadhyay *et al.*, (2006) that two proteins are likely to have the same function if their interaction partners are conserved.

$$\mathrm{SD}(p_i, p_j) = 1 - \frac{|N(p_i) \cap N(p_j)|}{2|N(p_i) \cup N(p_j)|} \tag{4.5}$$

$$\mathrm{Jc}(p_i, p_j) = 1 - \frac{|N(p_i) \cap N(p_j)|}{|N(p_i) \cup N(p_j)|} \tag{4.6}$$

where $N(p)$ represents adjacent neighbors of p in the network. The Srensen-Dice (i.e. Czekanovski–Dice) dissimilarity previously appeared in PRODISTIN (Brun *et al.*, 2003) for clustering of proteins and interactions.

4.1.3 Multi-dimensional scaling (MDS)

In order to apply discriminant analysis, we perform multi-dimensional scaling (MDS) (Mardia, 1978) to transform a distance matrix into feature vectors in a low-dimensional space. Given a double-centered distance matrix Δ, decompose it into eigenvectors V and a matrix of eigenvalues at its diagonal, the MDS variables X_l of the first l axes are chosen by minimizing square error.

$$X_l = V_l \Lambda_l^{1/2} R$$
$$\text{minimize}\,\mathrm{tr}[(\Delta^* - X_l X_l^T)^2]\,\text{subject to}\,RR^T = I$$

4.1.4 Multi-label Canonical discriminant analysis (MCDA)

4.1.4.1 Classic CDA

Given m population groups $g_1, g_2, \ldots, g_m$ and n variables of observed features $x_1, x_2, \ldots, x_n$ each of dimension p, CDA finds a transformation matrix C of canonical coefficients so that $Z = XC^T$, where X represents the feature variables x_i of dimension p and Z represents the transformed canonical variables z_i of dimension q (Seal, 1964). The transformation matrix C is defined to be the one that maximizes

$$C\Sigma_B C^T \qquad \text{subject to} \qquad C\Sigma_W C^T = I$$

and

$$(C\Sigma_W^{\frac{1}{2}})\Sigma_W^{-\frac{1}{2}}\Sigma_B\Sigma_W^{-\frac{1}{2}}(C\Sigma_W^{\frac{1}{2}})^T = F\Sigma_W^{-\frac{1}{2}}\Sigma_B\Sigma_W^{-\frac{1}{2}}F^T \qquad \text{subject to} \qquad FF^T = I$$

where Σ_B is the between-class covariance matrix and Σ_W is the within-class covariance matrix. The common covariance matrix is $\Sigma = F\Sigma_W^{-\frac{1}{2}}F^T$ at feature space and $\Sigma_C = I$ at canonical space.

By decomposing $\Sigma_W^{-\frac{1}{2}}\Sigma_B\Sigma_W^{-\frac{1}{2}}$, eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_q$ and eigenvectors $f_1, f_2, \ldots, f_q$ are obtained. The canonical coefficients can be computed by $C = \Sigma_W^{-\frac{1}{2}}F$, and the canonical correlation for the lth axis is defined by $\rho_l = \sqrt{\lambda_l/(1+\lambda_l)}$, $l = 1..q$.

Instead of calculating Σ_W and Σ_B and solving $\Sigma_B\Sigma_W^{-1}$ directly like in LDA, CDA solves the multivariate analog of the F statistic $SS_E^{-\frac{1}{2}}SS_H SS_E^{-\frac{1}{2}}$, where SS_H and SS_E are the multivariate analogs of the sum of squares and can be estimated from MANOVA through the multivariate test of equal means. The relationship between sum of squares and covariance matrices is: $\Sigma_B = SS_H/(m-1)$ and $\Sigma_W = SS_E/(n-m)$.

Traditionally, CDA is used for multiclass classification, where every variable belongs to only one of the multiple groups. The function membership L is represented by a size $n \times 1$ vector Y in CDA. By assuming a multivariate general linear model (GLM) $X = Y^T\beta + \epsilon$ for the observations X, the null hypothesis can be written as $H_0 : H\beta = 0$, where H is the hypothesis matrix of 0's and 1's that results in equal means for groups (Friendly, 2007). The sum of squares for hypothesis SS_H is estimated by the following (Timm, 1975), where the linear coefficients $\hat{\beta}$ can be estimated by least-square estimation.

$$\hat{SS}_H = (H\hat{\beta})^T\Big(H(Y^TY)^{-1}(H)^T\Big)^{-1}(H\hat{\beta}) \tag{4.7}$$

The sum of squares for error SS_E is estimated by the following.

$$\hat{SS}_E = \hat{\epsilon}^T \hat{\epsilon} \tag{4.8}$$

$\hat{SS}_H$ and $\hat{SS}_E$ can be weighted by a vector of prior probabilities $q = \{q_1, q_2, \dots, q_m\}$ of the m population groups and become $\hat{SS}_H = (H\hat{\beta})^T \Big(H(Y^T qY)^{-1}(H)^T \Big)^{-1} (H\hat{\beta})$ and $\hat{SS}_E = \hat{\epsilon}^T q \hat{\epsilon}$.

After canonical coefficients are obtained, the posterior probability that the canonical variable z of a new observation x belongs to the group g_j can be computed by a normalized exponential function of the Mahalanobis distance Maha^2 and prior probabilities:

$$\hat{Pr}(g_j|z) = \bigg(q_j \exp\Big(-\frac{1}{2}\text{Maha}^2(z, \hat{\mu}_{C,j})\Big) \bigg) \bigg/ \sum_{k=1}^{m} \bigg(q_k \exp\Big(-\frac{1}{2}\text{Maha}^2(z, \hat{\mu}_{C,k})\Big) \bigg) \tag{4.9}$$

where $\text{Maha}^2(z, \hat{\mu}_{C,k}) = (z - \hat{\mu}_{C,k})^T \hat{\Sigma}_C^{-1} (z - \hat{\mu}_{C,k})$, and the canonical mean $\hat{\mu}_{C,j}$ is estimated by the centroid of group g_j at canonical space, with the assumption that $\hat{\Sigma}_C = I$.

4.1.4.2 Covariance matrix estimation for multi-label CDA

For multi-label classification, we represent the functional association L by a design matrix D, where $D_{ij} = 1$ indicates that x_i belongs to group g_j, $i = 1..n$, $j = 1..m$. When using the same multivariate general linear model (GLM) $X = D^T\beta + \epsilon$ for the observations X and functional labels D, the null hypothesis can be written as $H_0 : H^{(j)}\beta = 0$ for each group g_j, and the sum of squares for hypothesis $SS_H^{(j)}$ is estimated as follows.

$$\hat{SS}_H^{(j)} = (H^{(j)}\hat{\beta})^T \Big(H^{(j)}(D^T D)^{-1}(H^{(j)})^T \Big)^{-1} (H^{(j)}\hat{\beta}) \tag{4.10}$$

The previous approach, multi-label LDA (Wang *et al.*, 2010), also estimates class-wise (i.e. group-wise) covariance matrix $\Sigma_W^{(j)}$ and $\Sigma_B^{(j)}$, and a final covariance matrix is defined as the matrix sum over all classes. They prove that when applying the multi-label LDA to binary classification, the covariance matrices stay the same, either by direct calculation or by class-wise estimation. We follow the same intuition and estimate the multi-label sum of squares for hypothesis by:

$$\hat{SS}_H = \sum_{j=1}^{m} \hat{SS}_H^{(j)} \tag{4.11}$$

We estimate the multi-label sum of squares for error using the same Equation 4.8.

4.1.4.3 Two-step regularization for MCDA

The coefficients of the general linear model can be computed using least-square estimation as $\hat{\beta} = (D^T D)^{-1} D^T X$. Since functional labels are often correlated in D which results in a singularity problem for $(D^T D)^{-1}$, we apply ridge regression (Hoerl and Kennard, 1970) to regularize it, and the ridge coefficients are defined by $\hat{\beta}^{\text{ridge}} = (D^T D + \lambda^{\text{ridge}} I)^{-1} D^T X$. We choose the regularization parameter λ^{ridge} to minimize the generalized cross validation (GCV) error (Wahba, 1990). Consequently, the regularized sum of squares for hypothesis $(\hat{SS}_H^{\text{ridge}})^{(j)}$ for group g_j is:

$$(\hat{SS}_H^{\text{ridge}})^{(j)} = (H^{(j)} \hat{\beta}^{\text{ridge}})^T \left(H^{(j)} (D_r^T D_r)^{-1} (H^{(j)})^T \right)^{-1} (H^{(j)} \hat{\beta}^{\text{ridge}}) \tag{4.12}$$

where $(D_r^T D_r)^{-1} = (D^T D + \lambda^{\text{ridge}} I)^{-1}$. The inverse of design matrix product, $(D_r^T D_r)^{-1}$, can be estimated by a right-matrix division on $\hat{\beta}^{\text{ridge}}$. The final $\hat{SS}_H^{\text{ridge}}$ is calculated by Equation 4.11.

For sum of squares for error $\hat{SS}_E$, and we apply the regularization for covariance matrix proposed by Guo *et al.*, (2007) which modifies the sample correlation matrix $\hat{R} = \hat{d}^{-\frac{1}{2}}\hat{SS}_E\hat{d}^{-\frac{1}{2}}$, where $\hat{d} = \text{diag}(\hat{SS}_E)$, into $\hat{R}_r = \lambda\hat{R} + (1-\lambda)I$, $0 \leq \lambda \leq 1$. The regularized sum of squares for error $(\hat{SS}_E)_r$ is computed by:

$$(\hat{SS}_E)_r = \hat{d}^{\frac{1}{2}}\hat{R}_r\hat{d}^{\frac{1}{2}} \tag{4.13}$$

We choose the regularization parameter λ to maximize the first canonical correlation ρ_1 calculated from eigen-decomposition of $(\hat{SS}_E)_r^{-\frac{1}{2}}\hat{SS}_H^{\text{ridge}}(\hat{SS}_E)_r^{-\frac{1}{2}}$, since it often results in better prediction.

4.2 Results

4.2.1 Evaluation

Yeast is among one of the species that currently provides the most abundant genetic and protein-protein interaction data. For the purpose of comparisons, we used the same BioGRID interaction data (Stark *et al.*, 2006) and MIPS Functional Catalogue (FunCat) (Mewes *et al.*, 2004) as in Wang *et al.*, (2015) for classification of yeast proteins. The first level of FunCat was used, which includes 17 high-level protein functions. On average, each protein belongs to 2.39 FunCat categories.

We also utilized the Gene Ontology (GO) terms from the Yeast Slim created by the *Saccharomyces* Genome Database (SGD) (Cherry *et al.*, 1998), which is a high-level subset of Gene Ontology. In total, 91 GO terms were chosen after removing the terms that annotate fewer than 30 or more than 500 of the yeast proteins.

We followed Wang *et al.*, (2015) and removed proteins with no functional annotation or with only one interaction. For FunCat classification, there are 4292 proteins with 47492 genetic interactions and 53393 protein-protein interactions, and

98142 interactions when the two types of interactions are mixed together. For Yeast Slim classification, there are 3698 proteins with 43893 genetic interactions and 32446 protein-protein interactions, and 73512 interactions when the two types of interactions are mixed together. Protein sequences were retrieved from SGD (Cherry *et al.*, 1998).

To integrate different types of data, we consider a linear combination of sequence, genetic and protein-protein interaction distance matrices with weights. The dimension of MDS variables can also affect the performance of prediction (Table 4.1), where a increase of the dimension does not necessarily causes a better prediction. We choose the first 500 MDS variables for the MIPS FunCat dataset and 100 for the Yeast Slim dataset. For MCDA, we do linear searches over the ranges for the regularization parameters, $10^{-2} \leq \lambda^{\text{ridge}} \leq 10^{10}$ and $0 \leq \lambda \leq 1$, and determine the best values that minimize the ridge regression error and that maximize the first canonical correlation, respectively. For the Yeast Slim dataset, MCDA is applied twice to further separate the functional groups to obtain improved performance.

To assess the performance of our protein function prediction algorithm, we perform five-fold cross validation for FunCat and Yeast Slim datasets, respectively. As for multiclass prediction, the multi-label prediction is also evaluated class by class (i.e. group by group). For each class, we impose multiple thresholds on the posterior probabilities to determine correspondingly if the functional label should be predicted. For each class and each threshold, we evaluate the predictions with the best F1 score, $\max_{0\leq t\leq 1}\{\text{F1}_t\}=\max_{0\leq t\leq 1}\{2 \times (\text{precision}_t \times \text{recall}_t)/(\text{precision}_t + \text{recall}_t)\}$, and the area under the curve of receiver operating characteristic (AUC-ROC), which is composed of false positive rates and true positive rates (FPR_t, TPR_t), $0 \leq t \leq 1$. The best F1 scores and the AUC-ROC's are averaged over all the classes (i.e. groups). We report each performance value as a range within one standard error over five-fold predictions.

Table 4.1: Effects of choosing different number of MDS variables for regularized MCDA on the five-fold prediction performances of the *S. cerevisiae* integrated data over 17 MIPS FunCat level-1 categories. (a) Average of best F1 scores (%). (b) Average AUC-ROC (%).

#MDS variables	(a) Best F1 score	(b) AUC-ROC
10	35.36±0.51	69.20±0.42
20	37.58±0.52	70.97±0.42
50	42.47±0.75	73.40±0.52
100	45.58±0.77	74.36±0.42
200	46.93±0.62	75.56±0.54
500	**48.85±0.55**	**77.02±0.38**
1000	47.73±0.60	75.57±0.46
2000	42.94±0.40	72.51±0.42

4.2.2 Predictions from protein sequences and interactions

Table 4.2 shows the prediction performances based on a combination of S&W sequence alignment distance, interaction partner dissimilarity of GI and PPI networks.

From MCDA, predictions made on interaction partner dissimilarity have better performances than S&W sequence alignment distances for both FunCat categories and Yeast Slim terms, with the protein-protein interaction (PPI) results better than the genetic interaction (GI) results. A similar trend is observed from the average of alignment distance and interaction dissimilarity, with the results of combining sequence and PPI matrices better than the results of combining sequence and GI matrices.

When predicting based solely on interaction dissimilarity in a PPI, GI or mixed network, Jaccard distance and Srensen-Dice dissimilarity performs similarly, but when predicting based on the average of alignment distance and interaction dissimilarity, Srensen-Dice dissimilarity performs better than with Jaccard distance.

While the performance of MDKC on Kullback-Leibler divergence of amino acid distributions is significantly better than MCDA on S&W sequence alignment distances, MCDA has better performance based on dissimilarity of interaction partners than MDKC on topological similarity. When sequence and interaction distances are combined, MCDA has better performance than MDKC.

While MDKC was tested on mixed GI and PPI networks, we tried two options: using one single distance matrix representing mixed GI and PPI networks, or computing and combining two distance matrices for GI and PPI networks. Predicting based on the mixed network produces better results than a single type of network probably due to more information included. When combining with the S&W sequence alignment distances, separate GI and PPI distance matrices are preferred than a mixed GI and PPI matrix for both FunCat categories and Yeast Slim terms. However, better AUC-ROC is obtained for a mixed GI and PPI matrix over the Yeast Slim terms.

Table 4.2: Comparison of distance/similarity features computed from *S. cerevisiae* protein sequences and interaction networks, and five-fold prediction performances of regularized MCDA and MDKC. (a) Average of best F1 scores over 17 MIPS FunCat level-1 categories (%). (b) Average AUC-ROC over 17 MIPS FunCat level-1 categories (%). (c) Average of best F1 scores over 91 Yeast Slim GO terms (%). (d) Average AUC-ROC over 91 Yeast Slim GO terms (%).

	MIPS FunCat		SGD Yeast Slim	
Distance/similarity	(a) F1 score	(b) AUC-ROC	(c) F1 score	(d) AUC-ROC
SW	33.11±0.33	62.47±0.30	16.21±0.14	61.08±0.35
KL (by MDKC)	**42.17**[1]			
SD-GI	**38.83±0.86**	69.94±0.56	30.85±0.52	76.47±0.44
Jc-GI	38.73±0.90	**70.25±0.61**		
SD-GI-1hop	35.22±0.32	66.31±0.33		
SD-PPI	41.70±0.39	72.52±0.34	36.61±0.57	77.62±0.39
Jc-PPI	**42.38±0.53**	**72.96±0.28**		
SD-PPI-1hop	37.55±0.28	69.84±0.64		
SD-GI/PPI	45.27±0.74	75.07±0.62	39.48±0.48	81.78±0.45
Jc-GI/PPI	**45.75±0.64**	**75.22±0.63**		
TM-GI/PPI (by MDKC)	40.03[1,2]			
SW + SD-GI[3]	**43.40±0.58**	**72.85±0.59**	32.31±0.49	78.22±0.64
SW + Jc-GI[3]	42.42±0.46	72.23±0.47		
SW + SD-PPI[3]	**43.75±0.56**	**73.68±0.12**	36.98±0.56	78.52±0.34
SW + Jc-PPI[3]	42.91±0.46	73.26±0.12		
SW + SD-GI/PPI[3]	47.04±0.34	75.85±0.42	40.35±0.43	**82.33±0.54**
SW + Jc-GI/PPI[3]	45.90±0.44	74.97±0.22		
SW + SD-GI + SD-PPI[4]	**49.50±0.42**	**77.00±0.51**	**40.92±0.37**	81.28±0.49
SW + Jc-GI + Jc-PPI[4]	46.93±0.39	76.18±0.26		
KL + SD-GI/PPI[4] (by MDKC)	≤45[5]		40.11[1,6]	

SW: One minus log odd score from Smith-Waterman sequence alignment

KL: Kullback-Leibler divergence of 3-amino-acid distributions

SD: Srensen-Dice dissimilarity

Jc: Jaccard distance

GI: Genetic interactions

PPI: Protein-protein interactions

GI/PPI: Mix of genetic and protein-protein interactions

1hop: Interaction partners at one hop distance are included

TM: Topological Measurement similarity (Pei and Zhang, 2005)

1: Reported by Wang *et al.*, (2015)

2: Prediction by randomly splitting the data into two halves (Wang *et al.*, 2015)

3: Average of sequence and interaction distance matrices

4: Linear combination of sequence and interaction distance matrices with weights

5: Performance estimated from Figure 3b in Wang *et al.*, (2015) for MDKC and other algorithms

6: Classification over 90 random GO terms across three GO domains (Wang *et al.*, 2015)

4.2.3 Effects of regularization and MCDA

Table 4.3 shows the performances from different versions of MCDA. With each step of the two-step regularization, the best F1 score improves further. The use of prior information of protein function distribution in MCDA does not give improved performance. Wang *et al.*, (2013) introduced function correlation to the multi-label protein function prediction problem. We applied their cosine product to the design matrix before running MCDA, but the performance drops significantly. Thus the cosine function-function correlation is not a suitable measure for MCDA.

Table 4.3: Five-fold prediction performances of regularized and non-regularized MCDA on *S. cerevisiae* integrated data over 17 MIPS FunCat level-1 categories. (a) Average of best F1 scores (%). (b) Average AUC-ROC (%).

RegH	RegE	Prior	Corr	(a) Best F1 score	(b) AUC-ROC
				47.18	75.04
×				47.43	75.43
	×			49.00	76.48
×	×			**49.50**	**77.00**
×	×	×		45.87	73.76
×	×		×	37.96	67.54

RegH: Ridge regression and regularization of sum of squares for hypothesis SS_H
RegE: Regularization of sum of squares for error SS_E
Prior: Prior of function distribution
Corr: Cosine function-function correlation (Wang *et al.*, 2015)

4.2.4 Performances by category

In this section, we take a look at regularized MCDA performance category by category. Table 4.4 shows the distribution of proteins in each FunCat group and the corresponding prediction performance. Figure 4.1 shows the degree of protein member overlap between FunCat categories.

Regularized MCDA outperforms MDKC or performs similarly for most FunCat categories, especially the ones with more than 500 proteins, with the exception of FunCat 18 (regulation of metabolism), 38 (transposable elements), which have fewer proteins but better performances, and the exception of 42 (biogenesis of cellular components), which has lot of proteins but only a comparable performance. We notice that the FunCat 38 has a significant low coverage, and the FunCat 42 includes significant parts of proteins from FunCat 40 and 43.

Regularized MCDA receives F1 scores $> 60\%$ for 6 categories, and scores 50-60% for 2 categories, and while MDKC has the scores for 4 and 1 categories.

MDKC only outperforms regularized MCDA at FunCat 34 (interaction with the environment) and FunCat 41 (development). The MCDA performance is possibly affected by the fact that most of proteins in 41 appear in 34, as shown in Figure 4.1. When we further check each of the dark spots in Figure 4.1, for most of these categories, MCDA shows only comparable performances and requires improvement.

MCDA receives a score $< 40\%$ for FunCat 18, 32, 34 and 41, for which we cannot find a reason related to the protein distribution or protein overlap in the functional categories. Thus, it is possibly associated with the classification ability of the observed features.

Table 4.4: Average of best F1 scores (%) of five-fold prediction on *S. cerevisiae* integrated data over 17 MIPS FunCat level-1 categories by (a) regularized MCDA; (b) MDKC and other algorithms in Wang *et al.*, (2015).

ID	Description	#proteins	(a) MCDA	(b) MDKC[1]
01	metabolism	1390	**60.14**	≤40
02	energy	327	44.53	≤45
10	cell cycle and DNA processing	965	**62.85**	≤∼60
11	transcription	995	**68.37**	≤60
12	protein synthesis	460	71.52	≤70
14	protein fate	1106	**53.33**	≤40
16	protein with binding function	1008	**44.15**	≤30
18	regulation of metabolism	240	**32.41**	≤20
20	cellular transport	974	**63.47**	≤∼60
30	cellular communication	230	51.17	≤∼50
32	cell rescue, defense and virulence	509	**38.37**	≤∼30
34	interaction with the environment	443	33.45	≤∼**45**
38	transposable elements	29	**65.59**	≤∼45
40	cell fate	264	44.08	≤∼45
41	development	66	16.14	≤**30**
42	biogenesis of cellular components	822	45.79	≤∼45
43	cell type differentiation	430	46.16	≤∼45
All		10258	**49.50**	≤45

1: Performances estimated from Figure 3b in Wang *et al.*, (2015)

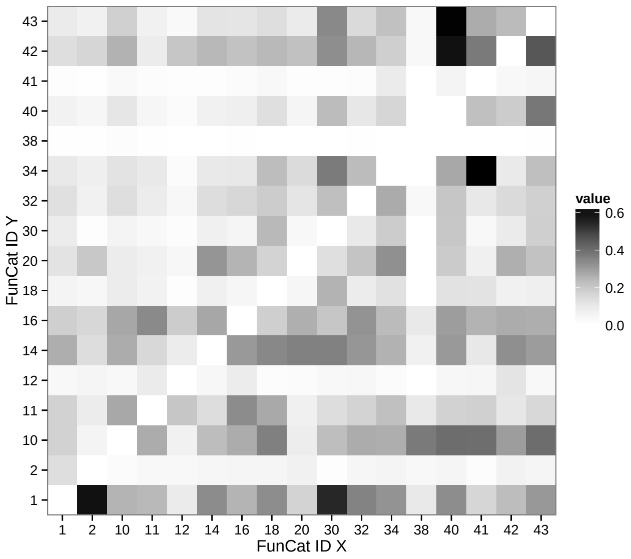
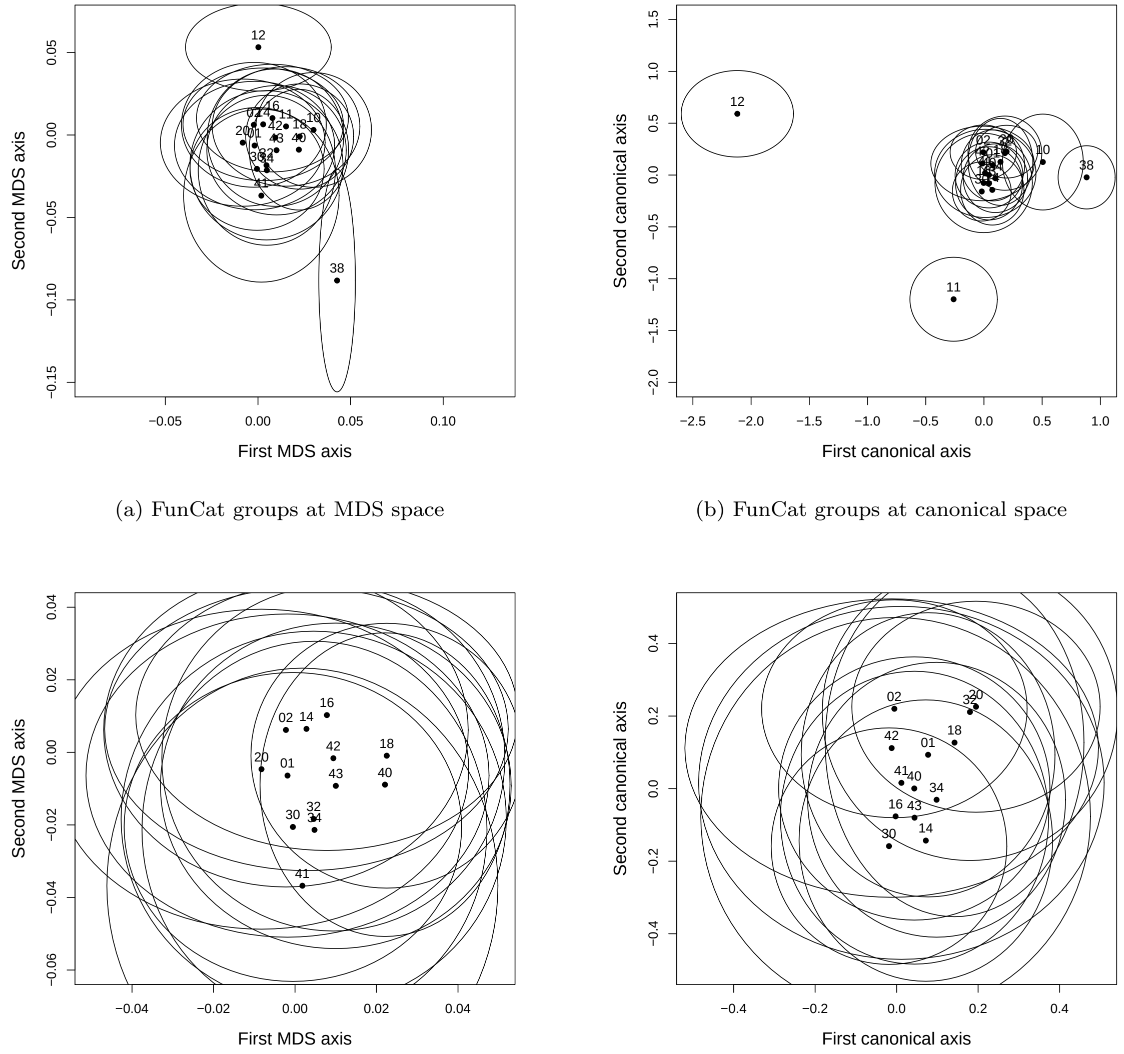


Figure 4.1: Protein overlap between category X and category Y relative to the size of category X among MIPS FunCat level-1 categories (%). Statistics are not shown for identical category pairs at the diagonal from the lower left to upper right.

Through dimension reduction of MDS, protein groups can be drawn on a two-dimensional plot that retains the original distance information. When the MDS variables are transformed with the canonical coefficients, groups are further separated on canonical axes. Figure 4.2 shows that on the first two axes of canonical space, FunCat groups 10, 11, 12 and 38 stand out when compared to the same groups at MDS space, and MCDA outperforms MDKC in these categories. FunCat 38 is among one of the outlying groups that MCDA outperforms the other approaches the most (this is also true for FunCat 01).

When looking closely at Figure 4.2, while FunCat 01 and 14 are not separated on the first axis at canonical space, they are well separated on the second axis and are predicted well. Also note that the centroids of FunCat 34 and 41 are close to the middle region with respect to the other groups at canonical space. Since the two categories have F1 scores smaller than 35%, the member variables are probably not separated on subsequent axes, either.



(a) FunCat groups at MDS space

(b) FunCat groups at canonical space

(c) Partial FunCat groups at MDS space

(d) Partial FunCat groups at canonical space

Figure 4.2: Visualization of 17 FunCat groups on *S. cerevisiae* integrated data, showing group centroids and 90% confidence circle on the first two axes of (a) MDS feature space; (b) MCDA canonical space; (c) MDS feature space showing a subset of categories that exclude groups 10, 11, 12 and 38; (d) MCDA canonical space showing a subset of categories that exclude groups 10, 11, 12 and 38. (The 90% confidence circle around group centroid k has a diameter$1.645 * s_l/\sqrt{n_k}$ along l-axis.)

Using MANOVA Wilks' test on normality, only the MDS variables of GI interactions that belong to or not belong to FunCat 38 (transposable elements) are Gaussian. Since many other FunCat groups with non-Gaussian variables are predicted well, MCDA demonstrates robustness to the violation of normality assumption.

5. CONCLUSIONS AND FUTURE WORK

We have presented two computational methods that help explore protein functions based on biomolecular interactions. The two methods contribute to the problem of local network alignment and functional module identification, and the problem of multi-label protein function prediction. They perform well respect to other software, including generating functionally enriched modules of interactions, and predicting functions more accurately for individual proteins.

5.1 Finding conserved functional modules using graphlet alignment of protein-protein interaction networks (GraphletAlign)

We have developed GraphletAlign, a method of local network alignment, that identifies graphlet alignments in protein-protein interaction networks. We show that it is possible to exhaustively enumerate all non-redundant graphlet alignments when the topology is small, and achieve a more complete coverage of proteins and protein interactions. By simple merges of overlapping graphlet alignments into conserved functional modules, the modules are shown to have comparable or higher sensitivity and specificity of functional enrichment with respect to other state-of-art software. Although only conserved graphlets of fewer than 5 proteins are enumerated, sensitivity stays relatively constant when maximum size of topology increases. Although our algorithm is slower than previous algorithms, its running time is approximately linear in the number of graphlet alignments that are generated. Among the algorithms which we have tested against, our algorithm is the only one that can identify alignments within a single species.

Our algorithm has the flexibility to use arbitrary sets of matching proteins or genes other than homologous genes, such as orthologous genes from COG (Tatusov *et al.*, 1997) or Inparanoid (Remm *et al.*, 2001), functionally similar genes (Li *et al.*, 2005), or genes with similar phylogenetic profiles (Pellegrini *et al.*, 1999). A different functional classification schemes can be used to annotate functional modules, such as KEGG Orthology (Kanehisa *et al.*, 2004).

5.1.1 Contributions

GraphletAlign, which is based on a discrete branch-and-bound algorithm, generates conserved functional modules of interactions covering generally more proteins. GraphletAlign can also be applied to single species other than just cross-species analyses.

5.1.2 Future work

GraphletAlign consists of two parts, including the phase of graphlet alignment and the phase of alignment join into functional modules. Improvement is needed to generate and prioritize the outputted functional modules. The concept of graphlet was originally developed to describe the distribution of linkages in a protein-protein interaction network. We have extended the concept to "conserved graphlets" in multiple networks. We may design a new scoring scheme for graphlet alignments by comparing the distribution of the conserved graphlets of different topologies to a null model. With the ranking based on scores, graphlet alignments can be chosen and developed to better-conserved functional modules. The redundancy between functional modules can also be reduced by score-based selection.

5.2 Protein function prediction from genetic and protein-protein interactions via regularized multi-label canonical discriminant analysis (MCDA)

We have developed a generalized version of canonical discriminant analysis that solves the multi-label problem in protein function prediction. We show that under the assumption that shared interaction partners lead to conserved functions, by choosing the associated distance measure, Srensen-Dice dissimilarity, for the proteins interactions, and by applying dimension reduction techniques, multi-dimensional scaling (MDS) and multi-label canonical discriminant analysis (MCDA), to transform the distance matrices into feature space, there can be a good separation for the protein variables from each functional group.

A linear combination of separate distance matrices from genetic and protein-protein interaction data enable better prediction than a merged interaction network does. By applying regularization to the two covariance matrices in MCDA, we solve the singularity problem and further improve the prediction performance. Regularized MCDA outperforms previous multi-label and multiclass single-label prediction approaches, especially for functional categories that contain a large number of proteins. On the other hand, the performance of MCDA can be sensitive to large protein overlap among functional groups and needs improvement. By visualizing MCDA canonical centroids of functional groups projected to selected two axes, it is possible to study group ranges and identify important outlying groups.

5.2.1 Contributions

Our method of multi-label protein function prediction focuses on three major aspects of the problem. The selected protein distance measure utilizes a hypothesis which links biomolecular interaction profiles to functions. The multi-label canonical discriminant analysis (MCDA) utilizes the related positions of the variables from each functional group and classify. The 2-step regularization applied to the MCDA solves the collinearity between functional labels or between protein features. These have not been emphasized by previous approaches.

5.2.2 Future work

The whole procedure actually provides a framework to solve the problem of multi-label protein function prediction. First, any distance measure, or called a kernel, of any type of protein data can be used for prediction and compared for performance. Utilizing MDS to transform a distance matrix to variables at feature space also enables application of any traditional multi-label classification algorithm to the problem.

The protein function prediction problem may be extended to multiple species. If the proteins of one species are better annotated, it might help to classify together the protein features from multiple species. The major challenge will be how to compare interaction neighborhood or other types of data across species and devise a valid distance measure.

We also hope to organize our implementation of MCDA and regularization into a R package and provide to other researchers.

Today, protein sequences of thousands of species are available in public databases, while interaction data of just over one hundred species have been identified and most of them are incomplete. With the advance of high-throughput experiments and availability of interaction data, we expect that automated functional analysis of proteome and comparison of interaction data can be of more and more importance. To compare patterns of interactions in interaction networks may seem more complicated than sequence comparison. However, using proteins as basic elements of comparison combined with network neighborhood analysis can be more efficient for functional studies, than comparing many more amino acids in whole-genome sequences. The direct association between the network model and cellular functions also provides better opportunities to understand biological processes.

REFERENCES

[1] R. Aebersold and M. Mann. Mass Spectrometry-based Proteomics. *Nature*, 422:198–207, 2003.

[2] Noga Alon, Phuong Dao, Iman Hajirasouliha, Fereydoun Hormozdiari, and S. Cenk Sahinalp. Biomolecular network motif counting and discovery by color coding. *Bioinformatics (Oxford, England)*, 24(13):i241–249, July 2008.

[3] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, October 1990.

[4] 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. *Nat. Genet.*, 25:25–29, 2000.

[5] 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. *Nat. Genet.*, 25:25–29, 2000. The Gene Ontology Consortium.

[6] J. S. Bader, A. Chaudhuri, J. M. Rothberg, and J. Chant. Gaining Confidence in High-throughput Protein Interaction Networks. *Nat. Biotechnol.*, 22:78–85, 2004.

[7] W. Bae, B. Xia, M. Inouye, and K. Severinov. *Escherichia coli CspA-family RNA chaperones are transcription antiterminators*, volume 97. 2000.

[8] S. Bandyopadhyay, R. Sharan, and T. Ideker. Systematic Identification of Functional Orthologs Based on Protein Network Comparison. *Genome Res.*, 16:428–435, 2006.

[9] S. Bandyopadhyay, R. Sharan, and T. Ideker. Systematic Identification of Functional Orthologs Based on Protein Network Comparison. *Genome Res.*, 16:428–435, 2006.

[10] Zafer Barutcuoglu, Robert E. Schapire, and Olga G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, April 2006.

[11] J. Berg and M. Lssig. Local Graph Alignment and Motif Search in Biological Networks. *Proc. Natl. Acad. Sci. USA*, 101:14689–14694, 2004.

[12] S. Bilke and C. Peterson. Topological Properties of Citation and Metabolic Networks. *Phys Rev E Stat Nonlin Soft Matter Phys.*, 64:036106, 2001.

[13] E. I. Boyle, S. Weng, J. Gollub, H. Jin, D. Botstein, J. M. Cherry, and G. Sherlock. GO::TermFinder-open Source Software for Accessing Gene Ontology Information and Finding Significantly Enriched Gene Ontology Terms Associated with a List of Genes. *Bioinformatics*, 20:3710–3715, 2004.

[14] Christine Brun, Franois Chevenet, David Martin, Jrme Wojcik, Alain Gunoche, and Bernard Jacq. Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. *Genome Biology*, 5(1):R6, December 2003.

[15] Mengfei Cao, Hao Zhang, Jisoo Park, Noah M. Daniels, Mark E. Crovella, Lenore J. Cowen, and Benjamin Hescott. Going the Distance for Protein Function Prediction: A New Distance Metric for Protein Interaction Networks. *PLoS ONE*, 8(10):e76339, October 2013.

[16] Andr C. P. L. F. de Carvalho and Alex A. Freitas. A Tutorial on Multi-label

Classification Techniques. In Ajith Abraham, Aboul-Ella Hassanien, and Vclav Snel, editors, *Foundations of Computational Intelligence Volume 5*, number 205 in Studies in Computational Intelligence, pages 177–195. Springer Berlin Heidelberg, 2009. DOI: 10.1007/978-3-642-01536-6_8.

[17] Nicol Cesa-Bianchi, Matteo Re, and Giorgio Valentini. Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference. *Machine Learning*, 88(1-2):209–241, 2012.

[18] J. M. Cherry, C. Adler, C. Ball, S. A. Chervitz, S. S. Dwight, E. T. Hester, Y. Jia, G. Juvik, T. Roe, M. Schroeder, S. Weng, and D. Botstein. SGD: Saccharomyces Genome Database. *Nucleic Acids Res.*, 26:73–79, 1998.

[19] B. Chor and T. Tuller. Biological Networks: Comparison, Conservation, and Evolution Via Relative Description Length. *J Comput Biol.*, 14:817–838, 2007.

[20] Domenico Cozzetto, Daniel WA Buchan, Kevin Bryson, and David T. Jones. Protein function prediction by massive integration of evolutionary analyses and multiple data sources. *BMC Bioinformatics*, 14(Suppl 3):S1, February 2013.

[21] T. Dandekar, S. Schuster, B. Snel, M. Huynen, and P. Bork. Pathway Alignment: Application to the Comparative Analysis of Glycolytic Enzymes. *Biochem, J.*, 1:115–124, 1999. 343 P.t.

[22] A. Delgado, S. Zaman, A. Muthaiyan, V. Nagarajan, M. O. Elasri, B. J. Wilkinson, and J. E. Gustafson. The Fusidic Acid Stimulon of Staphylococcus Aureus. *J. Antimicrob. Chemother.*, 62:1207–1214, 2008.

[23] Minghua Deng, Ting Chen, and Fengzhu Sun. An Integrated Probabilistic Model for Functional Prediction of Proteins. *Journal of Computational Biology*, 11(2-3):463–475, March 2004.

[24] Minghua Deng, Kui Zhang, Shipra Mehta, Ting Chen, and Fengzhu Sun. Prediction of Protein Function Using ProteinProtein Interaction Data. *Journal of*

Computational Biology, 10(6):947–960, December 2003.

[25] B. Dost, T. Shlomi, N. Gupta, E. Ruppin, V. Bafna, and R. Sharan. QNet: a Tool for Querying Protein Interaction Networks. *J. Comput. Biol.*, 15:913–925, 2008.

[26] J. Dutkowski and J. Tiuryn. Identification of Functional Modules from Conserved Ancestral Protein-protein Interactions. *Bioinformatics*, 23:I149–158, 2007.

[27] J. Espadaler, R. Aragues, N. Eswar, M. A. Marti-Renom, E. Querol, F. X. Aviles, A. Sali, and B. Oliva. Detecting Remotely Related Proteins by Their Interactions and Sequence Similarity. *Proc. Natl. Acad. Sci.*, 102:7151–7156, 2005.

[28] Tom Fawcett. An Introduction to ROC Analysis. *Pattern Recogn. Lett.*, 27(8):861–874, June 2006.

[29] S. Fields and O. Song. A Novel Genetic System to Detect Protein-protein Interactions. *Nature*, 340:245–246, 1989.

[30] J. Flannick, A. Novak, C. B. Do, B. S. Srinivasan, and S. Batzoglou. Automatic Parameter Learning for Multiple Local Network Alignment. *J. Comput. Biol.*, 16:1001–1022, 2009.

[31] J. Flannick, A. Novak, B. S. Srinivasan, H. H. McAdams, and S. Batzoglou. Grmlin: General and Robust Alignment of Multiple Large Interaction Networks. *Genome Res.*, 16:1169–1181, 2006.

[32] C. V. Forst and K. Schulten. Evolution of Metabolisms: A New Method for the Comparison of Metabolic Pathways Using Genomics Information. *J Comput Biol.*, 6:343–360, 1999.

[33] H. B. Fraser. Modularity and Evolutionary Constraint on Proteins. *Nat Genet.*, 37:351–352, 2005.

[34] H. B. Fraser, A. E. Hirsh, L. M. Steinmetz, C. Scharfe, and Feldman MW. Evolutionary Rate in the Protein Interaction Network. *Science*, 296:750–752, 2002.

[35] Michael Friendly. HE Plots for Multivariate Linear Models. *Journal of Computational and Graphical Statistics*, 16(2):421–444, June 2007.

[36] M. Gerstein, N. Lan, and R. Jansen. Proteomics Integrating interactomes. *Science*, 295:284–287, 2002. Comment.

[37] Eva Gibaja and Sebastin Ventura. A Tutorial on Multilabel Learning. *ACM Comput. Surv.*, 47(3):52:1–52:38, April 2015.

[38] L. Giot, J. S. Bader, C. Brouwer, A. Chaudhuri, B. Kuang, Y. Li, Y. L. Hao, C. E. Ooi, B. Godwin, E. Vitols, G. Vijayadamodar, P. Pochart, H. Machineni, M. Welsh, Y. Kong, B. Zerhusen, R. Malcolm, Z. Varrone, A. Collis, M. Minto, S. Burgess, L. McDaniel, E. Stimpson, F. Spriggs, J. Williams, K. Neurath, N. Ioime, M. Agee, E. Voss, K. Furtak, R. Renzulli, N. Aanensen, S. Carrolla, E. Bickelhaupt, Y. Lazovatsky, A. DaSilva, J. Zhong, C. A. Stanyon, White K. P. Finley RL, J.r., M. Braverman, T. Jarvie, S. Gold, M. Leach, J. Knight, R. A. Shimkets, M. P. McKenna, J. Chant, and J. M. Rothberg. A Protein Interaction Map of Drosophila Melanogaster. *Science*, 302:1727–1736, 2003.

[39] Vladimir Gligorijevi, Vuk Janji, and Nataa Prulj. Integration of molecular network data reconstructs Gene Ontology. *Bioinformatics*, 30(17):i594–i600, September 2014.

[40] J. A. Grochow and M. Kellis. Network Motif Discovery Using Subgraph Enumeration and Symmetry-breaking. *Lect. Notes Bioinformatics*, 4453:92–106, 2007.

[41] Yuanfang Guan, Chad L. Myers, David C. Hess, Zafer Barutcuoglu, Amy A. Caudy, and Olga G. Troyanskaya. Predicting gene function in a hierarchical

context with an ensemble of classifiers. *Genome Biology*, 9(Suppl 1):S3, June 2008.

[42] Xin Guo and Alexander J. Hartemink. Domain-oriented edge-based alignment of protein interaction networks. *Bioinformatics*, 25(12):i240–1246, June 2009.

[43] Y. Guo, T. Hastie, and R. Tibshirani. Regularized Linear Discriminant Analysis and Its Application in Microarrays. *Biostatistics*, 8:86–100, 2007.

[44] M. C. Gustin, J. Albertyn, M. Alexander, and K. Davenport. MAP Kinase Pathways in the Yeast Saccharomyces Cerevisiae. *Microbiol. Mol. Biol. Rev.*, 62:1264–1300, 1998.

[45] H. Hermjakob, L. Montecchi-Palazzi, C. Lewington, S. Mudali, S. Kerrien, S. Orchard, M. Vingron, B. Roechert, P. Roepstorff, A. Valencia, H. Margalit, J. Armstrong, A. Bairoch, G. Cesareni, D. Sherman, and R. Apweiler. IntAct: an Open Source Molecular Interaction Database. *Nucleic Acids Res.*, 32:D452–455, 2004.

[46] Haretsugu Hishigaki, Kenta Nakai, Toshihide Ono, Akira Tanigami, and Toshihisa Takagi. Assessment of prediction accuracy of protein function from proteinprotein interaction data. *Yeast*, 18(6):523–531, April 2001.

[47] A. E. Hoerl and R. W. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics*, 12:55–67, 1970.

[48] R. Hoffmann and A. Valencia. Protein Interaction: Same Network, Different Hubs. *Trends Genet*, 19:681–683, 2003.

[49] Pingzhao Hu, Sarath Chandra Janga, Mohan Babu, J. Javier Daz-Meja, Gareth Butland, Wenhong Yang, Oxana Pogoutse, Xinghua Guo, Sadhna Phanse, Peter Wong, Shamanta Chandran, Constantine Christopoulos, Anaies Nazarians-Armavil, Negin Karimi Nasseri, Gabriel Musso, Mehrab Ali, Nazila Nazemof, Veronika Eroukova, Ashkan Golshani, Alberto Paccanaro, Jack F Greenblatt,

Gabriel Moreno-Hagelsieb, and Andrew Emili. Global Functional Atlas of Escherichia coli Encompassing Previously Uncharacterized Proteins. *PLoS Biol*, 7(4):e1000096, April 2009.

[50] Ranjana Jaiswara, Diptarup Nandi, and Rohini Balakrishnan. Examining the Effectiveness of Discriminant Function Analysis and Cluster Analysis in Species Identification of Male Field Crickets Based on Their Calling Songs. *PLoS ONE*, 8(9):e75930, 2013.

[51] R. Jansen, H. Y.u., 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*, 302:449–453, 2003. Evaluation Studies.

[52] L. J. Jensen, M. Kuhn, M. Stark, S. Chaffron, C. Creevey, J. Muller, T. Doerks, P. Julien, A. Roth, M. Simonovic, P. Bork, and C. von Mering. STRING 8a Global View on Proteins and Their Functional Interactions in 630 Organisms. *Nucleic Acids Res.*, 37(Database issue):D412–416, 2009.

[53] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and Barabasi AL. The Large-scale Organization of Metabolic Networks. *Nature*, 407:651–654, 2000.

[54] R. Jiang, Z. Tu, T. Chen, and F. Sun. Network Motif Identification in Stochastic Networks. *Proc. Natl. Acad. Sci. USA*, 103:9404–9409, 2006.

[55] Trupti Joshi, Yu Chen, Jeffrey M. Becker, Nickolai Alexandrov, and Dong Xu. Genome-scale gene function prediction using multiple sources of high-throughput data in yeast Saccharomyces cerevisiae. *Omics: A Journal of Integrative Biology*, 8(4):322–333, 2004.

[56] M. P. Joy, A. Brock, D. E. Ingber, and S. Huang. High-betweenness Proteins in the Yeast Protein Interaction Network. *J Biomed Biotechnol*, 2005:96–103, 2005.

[57] M. Kalaev, V. Bafna, and R. Sharan. Fast and Accurate Alignment of Multiple Protein Networks. *J. Comput. Biol.*, 16:989–999, 2009.

[58] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, and Y. Yamanishi. KEGG for Linking Genomes to Life and the Environment. *Nucleic Acids Res.*, 36:D480–484, 2008.

[59] M. Kanehisa, S. Goto, S. Kawashima, Y. Okuno, and M. Hattori. The KEGG Resource for Deciphering the Genome. *Nucleic Acids Res.*, 32:D277–280, 2004.

[60] M. Kanehisa, S. Goto, S. Kawashima, Y. Okuno, and M. Hattori. The KEGG Resource for Deciphering the Genome. *Nucleic Acids Res.*, 32:D277–280, 2004.

[61] Ulas Karaoz, T. M. Murali, Stan Letovsky, Yu Zheng, Chunming Ding, Charles R. Cantor, and Simon Kasif. Whole-genome annotation by using evidence integration in functional-linkage networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2888–2893, March 2004.

[62] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. Efficient Sampling Algorithm for Estimating Subgraph Concentrations and Detecting Network Motifs. *Bioinformatics*, 20:1746–1758, 2004.

[63] 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. *Proc. Natl. Acad. Sci. USA*, 100:11394–11399, 2003.

[64] S. Kerrien, Y. Alam-Faruque, B. Aranda, I. Bancarz, A. Bridge, C. Derow, E. Dimmer, M. Feuermann, A. Friedrichsen, R. Huntley, C. Kohler, J. Khadake, C. Leroy, A. Liban, C. Lieftink, L. Montecchi-Palazzi, S. Orchard, J. Risse, K. Robbe, Thorneycroft D. Roechert, B., Y. Zhang, R. Apweiler, and H. Her-

mjakob. IntAct-open Source Resource for Molecular Interaction Data. *Nucleic Acids Res.*, 35:561–565, 2007.

[65] Wan Kyu Kim, Chase Krumpelman, and Edward M. Marcotte. Inferring mouse gene functions from genomic-scale data using a combined functional network/classification strategy. *Genome Biology*, 9 Suppl 1:S5, 2008.

[66] Yiannis A. I. Kourmpetis, Aalt D. J. van Dijk, Roeland C. H. J. van Ham, and Cajo J. F. ter Braak. Genome-Wide Computational Function Prediction of Arabidopsis Proteins by Integration of Multiple Data Sources. *Plant Physiology*, 155(1):271–281, January 2011.

[67] M. Koyutrk, A. Grama, and W. Szpankowski. An Efficient Algorithm for Detecting Frequent Subgraphs in Biological Networks. *Bioinformatics*, 20:SI200–207, 2004.

[68] Mehmet Koyutrk, Yohan Kim, Umut Topkara, Shankar Subramaniam, Wojciech Szpankowski, and Ananth Grama. Pairwise Alignment of Protein Interaction Networks. *Journal of Computational Biology*, 13(2):182–199, March 2006.

[69] O. Kuchaiev and N. Prulj. Integrative Network Alignment Reveals Large Regions of Global Network Similarity in Yeast and Human. *Bioinformatics*, 27:1390–1396, 2011.

[70] Deng M. Lanckriet, G. R.G., N. Cristianini, M. I. Jordan, and W. S. Noble. Kernel-based Data Fusion and Its Application to Protein Function Prediction in Yeast. *Pac. Symp. Biocomput.*, 9:300–311, 2004.

[71] Gert R. G. Lanckriet, Tijl De Bie, Nello Cristianini, Michael I. Jordan, and William Stafford Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, November 2004.

[72] Hyunju Lee, Zhidong Tu, Minghua Deng, Fengzhu Sun, and Ting Chen. Diffu-

sion Kernel-Based Logistic Regression Models for Protein Function Prediction. *OMICS: A Journal of Integrative Biology*, 10(1):40–55, March 2006.

[73] I. Lee, S. V. Date, A. T. Adai, and E. M. Marcotte. A Probabilistic Functional Network of Yeast Genes. *Science*, 306:1555–1558, 2004.

[74] B. Lemos, C. D. Meiklejohn, and D. L. Hartl. Regulatory Evolution Across the Protein Interaction Network. *Nat Genet.*, 36:1059–1060, 2004.

[75] M. Leone and A. Pagnani. Predicting Protein Functions with Message Passing Algorithms. *Bioinformatics*, 21:239–247, 2005.

[76] Michele Leone and Andrea Pagnani. Predicting protein functions with message passing algorithms. *Bioinformatics*, 21(2):239–247, January 2005.

[77] S. Letovsky and S. Kasif. Predicting Protein Function from Protein/protein Interaction Data: A Probabilistic Approach. *Bioinformatics*, 19:i197–204, 2003. (Suppl. 1).

[78] H. Li, M. Pellegrini, and D. Eisenberg. Detection of Parallel Functional Modules by Comparative Analysis of Genome Sequences. *Nat. Biotechnol.*, 23:253–260, 2005.

[79] S. Li, C. M. Armstrong, N. Bertin, H. Ge, S. Milstein, M. Boxem, P. O. Vidalain, J. D. Han, A. Chesneau, T. Hao, D. S. Goldberg, N. Li, M. Martinez, J. F. Rual, P. Lamesch, L. Xu, M. Tewari, S. L. Wong, L. V. Zhang, G. F. Berriz, L. Jacotot, P. Vaglio, J. Reboul, T. Hirozane-Kishikawa, Q. Li, H. W. Gabel, A. Elewa, B. Baumgartner, D. J. Rose, H. Yu, S. Bosak, R. Sequerra, A. Fraser, S. E. Mango, W. M. Saxton, S. Strome, S. Van Den Heuvel, F. Piano, J. Vandenhaute, C. Sardet, M. Gerstein, L. Doucette-Stamm, K. C. Gunsalus, J. W. Harper, M. E. Cusick, F. P. Roth, D. E. Hill, and M. Vidal. A Map of the Interactome Network of the Metazoan, C elegans. *Science*, 303:540–543, 2004.

[80] Zhenping Li, Shihua Zhang, Yong Wang, Xiang-Sun Zhang, and Luonan Chen. Alignment of molecular networks by integer quadratic programming. *Bioinformatics*, 23(13):1631–1639, July 2007.

[81] C. S. Liao, K. Lu, M. Baym, R. Singh, and B. Berger. IsoRankN: Spectral Methods for Global Alignment of Multiple Protein Networks. *Bioinformatics*, 25:i253–258, 2009.

[82] Hong-Wu Ma and An-Ping Zeng. The connectivity structure, giant strong component and centrality of metabolic networks. *Bioinformatics (Oxford, England)*, 19(11):1423–1430, July 2003.

[83] B. Mallet, F. Martos, L. Blambert, T. Pailler, and L. Humeau. Evidence for Isolation-by-habitat Among Populations of an Epiphytic Orchid Species on a Small Oceanic Island. *PLoS One*, 9:e87469, 2014.

[84] K. V. Mardia. Some Properties of Classical Multi-dimensional Scaling. *Commun. Statist. Theor. Meth.*, 7:1233–1241, 1978.

[85] S. Maslov and K. Sneppen. Specificity and Stability in Topology of Protein Networks. *Science*, 296:910–913, 2002.

[86] L. R. Matthews, P. Vaglio, J. Reboul, H. Ge, B. P. Davis, J. Garrels, S. Vincent, and M. Vidal. Identification of Potential Interaction Networks Using Sequence-based Searches for Conserved Protein-protein Interactions Or Interologs. *Genome Res.*, 11:2120–2126, 2001.

[87] B. D. McKay. Practical Graph Isomorphism. *Congressus Numerantium*, 30:45–87, 1981.

[88] B. D. McKay. Isomorph-free Exhaustive Generation. *J. Algorithms*, 26:306–324, 1998.

[89] H. W. Mewes, C. Amid, R. Arnold, D. Frishman, U. Gldener, G. Mannhaupt, M. Mnsterktter, P. Pagel, N. Strack, V. Stmpflen, J. Warfsmann, and

A. Ruepp. MIPS: Analysis and Annotation of Proteins from Whole Genomes. *Nucleic Acids Res.*, 32:D41–44, 2004.

[90] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 298:824–827, 2002.

[91] Sara Mostafavi, Debajyoti Ray, David Warde-Farley, Chris Grouios, and Quaid Morris. GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biology*, 9(Suppl 1):S4, 2008.

[92] E. Nabieva, K. Jim, A. Agarwal, B. Chazelle, and M. Singh. Whole-proteome Prediction of Protein Function Via Graph-theoretic Analysis of Interaction Maps. *Bioinformatics*, (Suppl. 1):i302–310, 2005.

[93] Elena Nabieva, Kam Jim, Amit Agarwal, Bernard Chazelle, and Mona Singh. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, 21(suppl 1):i302–i310, June 2005.

[94] W Noble and Asa Ben-Hur. Integrating information for protein function prediction. *Bioinformatics-From Genomes to Therapies*, 3:1297–1314, 2007.

[95] Guillaume Obozinski, Gert Lanckriet, Charles Grant, Michael I. Jordan, and William S. Noble. Consistent probabilistic outputs for protein function prediction. *Genome Biology*, 9(Suppl 1):S6, June 2008.

[96] P. Pagel, H. W. Mewes, and D. Frishman. Conservation of Protein-protein Interactions - Lessons from Ascomycota. *Trends Genet.*, 20:72–76, 2004. Review.

[97] L. Parida. Discovering Topological Motifs Using a Compact Notation. *J. Comput. Biol.*, 14:300–323, 2007.

[98] R. Pastor-Satorras, E. Smith, and R. V. Sole. Evolving Protein Interaction Networks Through Gene Duplication. *J Theor Biol.*, 222:199–210, 2003.

[99] Paul Pavlidis and Jesse Gillis. Progress and challenges in the computational

prediction of gene function using networks. *F1000Research*, 1:14, 2012.

[100] Lourdes Pea-Castillo, Murat Tasan, Chad L. Myers, Hyunju Lee, Trupti Joshi, Chao Zhang, Yuanfang Guan, Michele Leone, Andrea Pagnani, Wan K. Kim, Chase Krumpelman, Weidong Tian, Guillaume Obozinski, Yanjun Qi, Sara Mostafavi, Guan N. Lin, Gabriel F. Berriz, Francis D. Gibbons, Gert Lanckriet, Jian Qiu, Charles Grant, Zafer Barutcuoglu, David P. Hill, David Warde-Farley, Chris Grouios, Debajyoti Ray, Judith A. Blake, Minghua Deng, Michael I. Jordan, William S. Noble, Quaid Morris, Judith Klein-Seetharaman, Ziv Bar-Joseph, Ting Chen, Fengzhu Sun, Olga G. Troyanskaya, Edward M. Marcotte, Dong Xu, Timothy R. Hughes, and Frederick P. Roth. A critical assessment of Mus musculus gene function prediction using integrated genomic evidence. *Genome Biology*, 9(Suppl 1):S2, June 2008.

[101] P. Pei and A. Zhang. A Topological Measurement for Weighted Protein Interaction Network. *Proc. IEEE Comput. Syst. Bioinform. Conf.*, pages 268–278, 2005.

[102] M. Pellegrini, E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates. Assigning Protein Functions by Comparative Genome Analysis: Protein Phylogenetic Profiles. *Proc. Natl. Acad. Sci. USA*, 96:4285–4288, 1999.

[103] R. V. Pouyat, I. D. Yesilonis, J. Russell-Anelli, and N. K. Neerchal. Soil Chemical and Physical Properties That Differentiate Urban Land-use and Cover Types. *Soil Sci. Soc. Am. J.*, 71:1010–1019, 2007.

[104] N. Prulj, D. G. Corneil, and I. Jurisica. Efficient Estimation of Graphlet Frequency Distributions in Protein-protein Interaction Networks. *Bioinformatics*, 22:974–980, 2006.

[105] N. Przulj. Biological Network Comparison Using Graphlet Degree Distribution. *Bioinformatics*, 23:e177–183, 2007.

[106] H. Qin, H. H. Lu, W. B. Wu, and Li WH. Evolution of the Yeast Protein Interaction Network. *Proc Natl Acad Sci*, 100:12820–12824, 2003.

[107] Aebersold R and Mann M. Mass spectrometry-based proteomics. *Nature*, volume 422. 2003.

[108] Predrag Radivojac. A (not so) quick introduction to protein function prediction. 2013.

[109] Predrag Radivojac, Wyatt T. Clark, Tal Ronnen Oron, Alexandra M. Schnoes, Tobias Wittkop, Artem Sokolov, Kiley Graim, Christopher Funk, Karin Verspoor, Asa Ben-Hur, Gaurav Pandey, Jeffrey M. Yunes, Ameet S. Talwalkar, Susanna Repo, Michael L. Souza, Damiano Piovesan, Rita Casadio, Zheng Wang, Jianlin Cheng, Hai Fang, Julian Gough, Patrik Koskinen, Petri Trnen, Jussi Nokso-Koivisto, Liisa Holm, Domenico Cozzetto, Daniel W. A. Buchan, Kevin Bryson, David T. Jones, Bhakti Limaye, Harshal Inamdar, Avik Datta, Sunitha K. Manjari, Rajendra Joshi, Meghana Chitale, Daisuke Kihara, Andreas M. Lisewski, Serkan Erdin, Eric Venner, Olivier Lichtarge, Robert Rentzsch, Haixuan Yang, Alfonso E. Romero, Prajwal Bhat, Alberto Paccanaro, Tobias Hamp, Rebecca Kaner, Stefan Seemayer, Esmeralda Vicedo, Christian Schaefer, Dominik Achten, Florian Auer, Ariane Boehm, Tatjana Braun, Maximilian Hecht, Mark Heron, Peter Hnigschmid, Thomas A. Hopf, Stefanie Kaufmann, Michael Kiening, Denis Krompass, Cedric Landerer, Yannick Mahlich, Manfred Roos, Jari Bjrne, Tapio Salakoski, Andrew Wong, Hagit Shatkay, Fanny Gatzmann, Ingolf Sommer, Mark N. Wass, Michael J. E. Sternberg, Nives kunca, Fran Supek, Matko Bonjak, Pane Panov, Sao Deroski, Tomislav muc, Yiannis A. I. Kourmpetis, Aalt D. J. van Dijk, Cajo J. F. ter Braak, Yuanpeng Zhou, Qingtian Gong, Xinran Dong, Weidong Tian, Marco Falda, Paolo Fontana, Enrico Lavezzo, Barbara Di Camillo, Stefano Toppo,

Liang Lan, Nemanja Djuric, Yuhong Guo, Slobodan Vucetic, Amos Bairoch, Michal Linial, Patricia C. Babbitt, Steven E. Brenner, Christine Orengo, Burkhard Rost, Sean D. Mooney, and Iddo Friedberg. A large-scale evaluation of computational protein function prediction. *Nature Methods*, 10(3):221–227, March 2013.

[110] Matteo Re, Marco Mesiti, and Giorgio Valentini. A Fast Ranking Algorithm for Predicting Gene Functions in Biomolecular Networks. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 9(6):1812–1818, November 2012.

[111] M. Remm, C. E.V. Storm, and E. L.L. Sonnhammer. Automatic Clustering of Orthologs and In-paralogs from Pairwise Species Comparisons. *J. Mol. Biol.*, 314:1041–1052, 2001.

[112] B. Rost, J. Liu, R. Nair, K.O. Wrzeszczynski, and Y. Ofran. Automatic prediction of protein function. *Cellular and Molecular Life Sciences CMLS*, 60(12):2637–2650, 2003.

[113] J. F. Rual, K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, N. Li, G. F. Berriz, F. D. Gibbons, M. Dreze, N. Ayivi-Guedehoussou, N. Klitgord, C. Simon, M. Boxem, S. Milstein, J. Rosenberg, D. S. Goldberg, L. V. Zhang, S. L. Wong, G. Franklin, S. Li, J. S. Albala, J. Lim, C. Fraughton, E. Llamosas, S. Cevik, C. Bex, P. Lamesch, R. S. Sikorski, J. Vandenhaute, H. Y. Zoghbi, A. Smolyar, S. Bosak, R. Sequerra, L. Doucette-Stamm, M. E. Cusick, D. E. Hill, F. P. Roth, and M. Vidal. Towards a Proteome-scale Map of the Human Protein-protein Interaction Network. *Nature*, 437:1173–1178, 2005.

[114] M. R. Said, T. J. Begley, A. V. Oppenheim, D. A. Lauffenburger, and Samson LD. Global Network Analysis of Phenotypic Effects: Protein Networks and Toxicity Modulation in Saccharomyces Cerevisiae. *Proc Natl Acad Sci.*, 101:18006–18011, 2004.

[115] B. Schwikowski, P. Uetz, and S. Fields. A network of protein-protein interactions in yeast. *Nature Biotechnology*, 18(12):1257–1261, December 2000.

[116] H. L. Seal. *Multivariate Statistical Analysis for Biologists.* Methuen, 1964.

[117] R. Sharan, T. Ideker, B. Kelley, R. Shamir, and Karp RM. Identification of Protein Complexes by Comparative Analysis of Yeast and Bacterial Protein Interaction Data. *J Comput Biol.*, 12:835–846, 2005.

[118] 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. *Proc. Natl. Acad. Sci. USA*, 102:1974–1979, 2005.

[119] Roded Sharan, Igor Ulitsky, and Ron Shamir. Networkbased prediction of protein function. *Molecular Systems Biology*, 3(1):88, January 2007. Functional annotation of proteins is a fundamental problem in the postgenomic era. The recent availability of protein interaction networks for many model species has spurred on the development of computational methods for interpreting such data in order to elucidate protein function. In this review, we describe the current computational approaches for the task, including direct methods, which propagate functional information through the network, and moduleassisted methods, which infer functional modules within the network and use those for the annotation task. Although a broad variety of interesting approaches has been developed, further progress in the field will depend on systematic evaluation of the methods and their dissemination in the biological community. Mol Syst Biol. 3: 88.

[120] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network Motifs in the Transcriptional Regulation Network of Escherichia Coli. *Nat. Genet.*, 31:64–68, 2002.

[121] Hyunjung Shin, Andreas Martin Lisewski, and Olivier Lichtarge. Graph sharp-

ening plus graph integration: a synergy that improves protein functional classification. *Bioinformatics*, 23(23):3217–3224, December 2007.

[122] T. Shlomi, D. Segal, E. Ruppin, and R. Sharan. QPath: a Method for Querying Pathways in a Protein-protein Interaction Network. *BMC Bioinformatics*, 7:199, 2006.

[123] R. Singh, J. Xu, and B. Berger. Global Alignment of Multiple Protein Interaction Networks with Application to Functional Orthology Detection. *Proc. Natl. Acad. Sci. USA*, 105:12763–12768, 2008.

[124] T. F. Smith and M. S. Waterman. Identification of Common Molecular Subsequences. *J. Mol. Biol.*, 147:195–197, 1981.

[125] B. S. Srinivasan, A. F. Novak, J. A. Flannick, S. Batzoglou, and H. H. McAdams. Integrated Protein Interaction Networks for 11 Microbes. *Lect. Notes Bioinformatics*, 3909:1–14, 2006.

[126] C. Stark, Reguly T. Breitkreutz, B.-J., L. Boucher, A. Breitkreutz, and M. Tyers. BioGRID: a General Repository for Interaction Datasets. *Nucleic Acids Res.*, 34:D535–539, 2006.

[127] U. Stelzl, U. Worm, M. Lalowski, C. Haenig, F. H. Brembeck, H. Goehler, M. Stroedicke, M. Zenkner, A. Schoenherr, S. Koeppen, J. Timm, S. Mintzlaff, C. Abraham, N. Bock, S. Kietzmann, A. Goedde, E. Toksoz, A. Droege, S. Krobitsch, B. Korn, W. Birchmeier, H. Lehrach, and E. E. Wanker. A Human Protein-protein Interaction Network: A Resource for Annotating the Proteome. *Cell*, 122:830–832, 2005.

[128] S. Suthram, T. Sittler, and T. Ideker. The Plasmodium Protein Network Diverges from Those of Other Eukaryotes. *Nature*, 438:108–112, 2005.

[129] R. L. Tatusov, E. V. Koonin, and D. J. Lipman. A Genomic Perspective on Protein Families. *Science*, 278:631–637, 1997.

[130] Weidong Tian, Lan V. Zhang, Murat Taan, Francis D. Gibbons, Oliver D. King, Julie Park, Zeba Wunderlich, J. Michael Cherry, and Frederick P. Roth. Combining guilt-by-association and guilt-by-profiling to predict Saccharomyces cerevisiae gene function. *Genome Biology*, 9(Suppl 1):S7, June 2008.

[131] Y. Tian, R. C. McEachin, C. Santos, D. J. States, and J. M. Patel. SAGA: a Subgraph Matching Tool for Biological Graphs. *Bioinformatics*, 23:232–239, 2007.

[132] N. H. Timm. *Multivariate Analysis, with Applications in Education and Psychology.* Brooks/Cole Pub. Co., 1975.

[133] Y. Tohsato, H. Matsuda, and A. Hashimoto. A Multiple Alignment Algorithm for Metabolic Pathway Analysis Using Enzyme Hierarchy. *Proc Int Conf Intell Syst Mol Biol.*, 8:376–383, 2000.

[134] O. G. Troyanskaya, K. Dolinski, A. B. Owen, R. B. Altman, and D. Botstein. A Bayesian Framework for Combining Heterogeneous Data Sources for Gene Function Prediction (in Saccharomyces Cerevisiae). *Proc. Natl. Acad. Sci.*, 100:8348–8353, 2003.

[135] K. Tsuda and W. S. Noble. Learning Kernels from Biological Networks by Maximizing Entropy. *Bioinformatics*, (Suppl. 1):i326–333, 2004.

[136] Koji Tsuda, HyunJung Shin, and Bernhard Schlkopf. Fast protein classification with multiple networks. *Bioinformatics*, 21(suppl 2):ii59–ii65, January 2005.

[137] 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*, 403:623–627, 2000.

[138] J. R. Ullmann. An Algorithm for Subgraph Isomorphism. *Journal of the ACM*,

23:31–42, 1976.

[139] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani. Global Protein Function Prediction from Protein-protein Interaction Networks. *Nat. Biotechnol.*, 21:697–700, 2003.

[140] Alexei Vazquez, Alessandro Flammini, Amos Maritan, and Alessandro Vespignani. Global protein function prediction from protein-protein interaction networks. *Nature Biotechnology*, 21(6):697–700, June 2003.

[141] G. Wahba. *Spline Models for Observational Data*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, January 1990.

[142] A. J. Walhout, R. Sordella, X. Lu, J. L. Hartley, G. F. Temple, M. A. Brasch, N. Thierry-Mieg, and M. Vidal. Protein interaction mapping in C. elegans using proteins involved in vulval development. *Science (New York, N.Y.)*, 287(5450):116–122, January 2000.

[143] Hua Wang, Chris Ding, and Heng Huang. Multi-label Linear Discriminant Analysis. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision ECCV 2010*, number 6316 in Lecture Notes in Computer Science, pages 126–139. Springer Berlin Heidelberg, 2010.

[144] Hua Wang, Heng Huang, and Chris Ding. FunctionFunction Correlated Multi-label Protein Function Prediction over Interaction Networks. *Journal of Computational Biology*, 20(4):322–343, 2013.

[145] Hua Wang, Heng Huang, and Chris Ding. Correlated Protein Function Prediction via Maximization of Data-Knowledge Consistency. *Journal of Computational Biology*, 22(6):546–562, April 2015.

[146] Peggy I. Wang and Edward M. Marcotte. It's the machine that matters: Predicting gene function and phenotype from protein networks. *Journal of Pro-*

teomics, 73(11):2277 – 2289, 2010. Model organism proteomics.

[147] S. Wernicke. Efficient Detection of Network Motifs. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 3:347–359, 2006.

[148] Andrew Wong and Hagit Shatkay. Protein Function Prediction using Text-based Features extracted from the Biomedical Literature: The CAFA Challenge. *BMC Bioinformatics*, 14(Suppl 3):S14, 2013.

[149] S. L. Wong, L. V. Zhang, L.i. Z. Tong, A. H.Y., D. S. Goldberg, O. D. King, G. Lesage, M. Vidal, B. Andrews, H. Bussey, C. Boone, and F. P. Roth. Combining Biological Networks to Predict Genetic Interactions. *Proc. Natl. Acad. Sci.*, 101:15682–15687, 2004.

[150] Q. Wu X Liu and R. Jiang. Align Human Interactome with Phenome to Identify Causative Genes and Networks Underlying Disease Families. *Bioinformatics*, 25:98–104, 2009.

[151] S. Wuchty. Evolution and Topology in the Yeast Protein Interaction Network. *Genome Res.*, 14:1310–1314, 2004.

[152] S. Wuchty, Z. N. Oltvai, and Barabasi AL. Evolutionary Conservation of Motif Constituents in the Yeast Protein Interaction Network. *Nat Genet.*, 35:176–179, 2003.

[153] I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, and Eisenberg D. Kim, S.-M. DIP: the Database of Interacting Proteins: a Research Tool for Studying Cellular Networks of Protein Interactions. *Nucleic Acids Res.*, 30:303–305, 2002.

[154] Guoxian Yu, Carlotta Domeniconi, Huzefa Rangwala, Guoji Zhang, and Zhiwen Yu. Transductive Multi-label Ensemble Classification for Protein Function Prediction. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 1077–1085, New

York, NY, USA, 2012. ACM.

[155] Guoxian Yu, Hailong Zhu, Carlotta Domeniconi, and Maozu Guo. Integrating multiple networks for protein function prediction. *BMC Systems Biology*, 9(Suppl 1):S3, 2015.

[156] H. Yu, N. M. Luscombe, H. X. Lu, Y. Zhu X Xia, J. D. Han, N. Bertin, S. Chung, M. Vidal, and M. Gerstein. Annotation Transfer Between Genomes: Protein-protein Interologs and Protein-DNA Regulogs. *Genome Res.*, 14:1107–1118, 2004.

[157] M. Zaslavskiy, F. Bach, and Vert JP. Global Alignment of Protein-protein Interaction Networks by Graph Matching Methods. *Bioinformatics*, 25:i259–267, 2009.

[158] Xiao-Fei Zhang and Dao-Qing Dai. A Framework for Incorporating Functional Interrelationships into Protein Function Prediction Algorithms. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 9(3):740–753, May 2012.