Tel-Aviv University
School of Computer Science

Algorithms and tools for the alignment of multiple protein networks

This thesis is submitted in partial fulfillment
of the requirements toward the M.Sc. degree
Tel-Aviv University
School of Computer Science

by
Maxim Kalaev

The research work in this thesis has been carried out
under the supervision of Prof. Roded Sharan

July, 2008

Abstract

Comparative analysis of protein networks has proven to be a powerful approach for elucidating network structure and predicting protein function and interaction. A fundamental challenge for the success of this approach is the development of efficient algorithms and tools for comparing multiple networks. In this thesis we present two novel tools for comparative network analysis.

The first tool is a framework for the alignment of multiple networks. At the heart of the framework is a novel representation of multiple networks that is only linear in their size as opposed to current exponential representations. The alignment algorithm is very efficient, being capable of aligning 10 networks with tens of thousands of proteins each in minutes. We show that our algorithm outperforms previous strategies for the problem, and produces results that are more in line with current biological knowledge. A paper describing this framework appeared in the Proceedings of RECOMB 2008.

The second tool is a web-server implementing the NetworkBLAST algorithm for identifying protein complexes in protein-protein interaction networks. The server can analyze a single network or two networks from different species. In the latter case, NetworkBLAST outputs a set of putative complexes that are evolutionarily conserved across the two networks. The server accepts input data in various formats and provides a graphical visualization of the identified complexes. The application report appeared in Bioinformatics.

Contents

Acknowledgments

I would like to thank my supervisor Roded Sharan, for the initial idea and for the guidance throughout the research and development stages, Vineet Bafna from University of California, San Diego for his contribution to the fundamental theory behind this work and Israel Science Foundation for supporting this research (grant no. 385/06).

Chapter 1

Introduction

Modern technological advances enable the systematic characterization of protein-protein interaction (PPI) networks across multiple species. Procedures such as yeast two-hybrid [20] and protein co-immunoprecipitation [1] are routinely employed nowadays to generate large-scale protein interaction networks for human and most model species [11, 15, 16, 19, 40]. An arising challenge is to organize the accumulating network data into models of cellular machinery. As in other biological domains, a comparative approach provides a powerful basis for addressing this challenge, calling for algorithms for protein network alignment.

The network alignment problem calls for identifying network regions that are conserved in their sequence and interaction pattern across two or more species. While the general problem is hard, generalizing subgraph isomorphism, heuristic methods have been devised to tackle it. One heuristic approach for problem creates a merged representation of the networks being compared, called a network alignment graph, facilitating the search for conserved subnetworks. In a network alignment graph, the nodes represent sets of proteins, one from each species, and the edges represent conserved PPIs across the two species. The alignment may consist of one-to-one correspondence between proteins across the two networks; however, in general there may be a many-to-many correspondence between proteins. This scenario can occur, for instance, when a single protein from one species is homologous to multiple proteins from the other species.

The network alignment paradigm has been applied successfully by a number of authors to search for conserved pathways [25] and complexes [26, 35, 37]. However, its extension to more than a few networks proved difficult due to the

exponential growth of the alignment graph with the number of species. Recently, an algorithm was suggested to overcome this difficult, proposing the idea of imitating progressive sequence alignment techniques [17]. The latter algorithm was successfully applied to align up to 10 microbial networks. Recently, Dutkowsky and Tiuryn [9] proposed another framework for efficient alignment of multiple networks, but this approach was applied to date to three networks only.

In this thesis we develop two tools for network alignment. First, we present a new algorithm for multiple network alignment. The method is based on a novel representation of the network data, which replaces the network alignment graph representation, leading to dramatic reduction in time and memory consumption. We compare our algorithm to previous approaches on various data sets, showing that it is extremely fast and accurate, outperforming a recent method based on progressive alignment ideas.

Second, we describe a web-server application implementing the NetworkBLAST algorithm developed by Sharan et al. [35] for detecting proteins complexes that are conserved across species. The web-server allows uploading and processing of a single network or two networks from different species to search for putative complexes that are evolutionarily conserved across the two networks. Several formats for input data are supported, including PSI MI 2.5 XML as used by DIP [43]. In the latter case, interaction reliability scores are automatically assigned by the server, using a logistic regression model (see [35]). The server generates a graphical representation of the identified putative protein complexes, providing a simple and intuitive user interface.

The thesis is organized as follows: In the next chapter we provide a biological background describing the main methods that are used nowadays for detecting protein-protein interactions. In Chapter 3 we describe previous algorithms for multiple network alignment. In Chapter 4 we present a novel algorithm for multiple network alignment. Chapter 5 describes experimental results, validation methods and results of comparison to other methods. A web-server application for the NetworkBLAST algorithm is presented in Chapter 6. Finally, we provide a summary of the research in Chapter 7.

Chapter 2

Biological Background

In this chapter we give a biological background and review key large-scale technologies for detecting protein-protein interactions.

2.1 Protein Protein Interactions

Protein-protein interactions are biochemical reactions between proteins molecules. Interactions between the proteins are mandatory for virtually all living processes in the cell. For example, signal transduction within the cell is based on chains of protein interactions, many of which involve kinase enzymes, proteins which react with other proteins to modify their function, usually by adding to them a phosphate group (see Figure 2.1). Proteins can form also long-lasting interactions, creating *proteins complexes*. The complexes are stable over time, performing complicate functions requiring the collaboration of several proteins. For example, the proteasomes are large proteins complexes that serve as the protein degradation machinery within the cell, degrading unneeded or damaged proteins down to peptides, which are further split to proteins acids and reused in the synthesis of new proteins (see Figure 2.2).

2.2 Protein-Protein Interaction Detection

As protein interactions are so fundamental, dozens of methods were developed to detect them. Each method has its pros and cons, such as cost, rate of false negatives, false positives, scalability and time required for the experiment. The

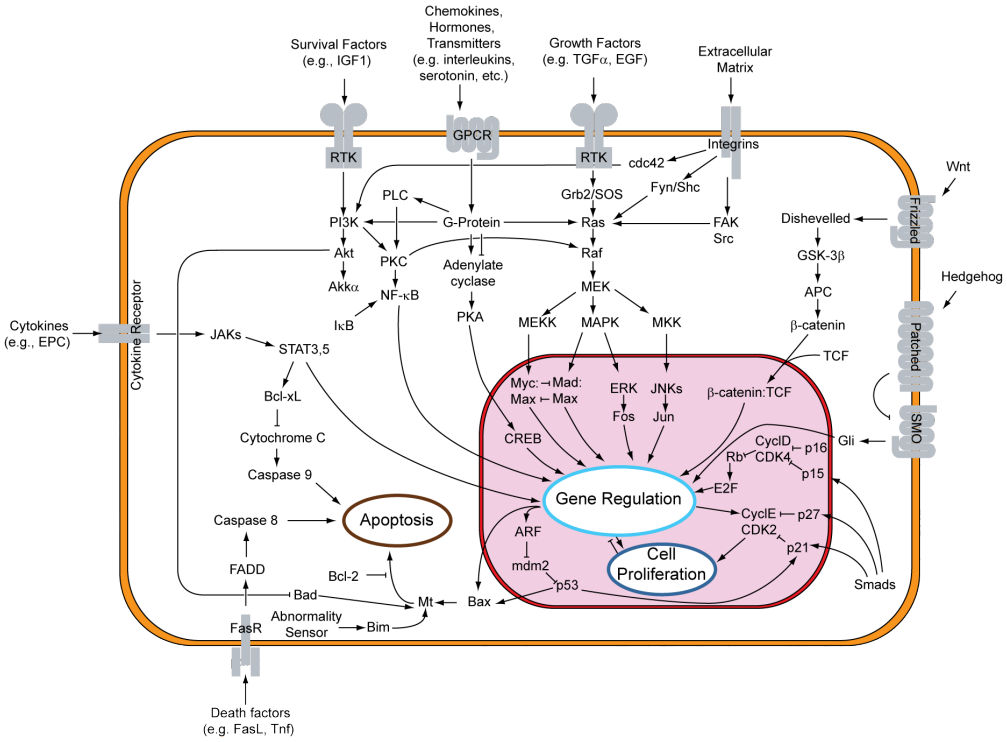

Figure 2.1: An illustration of signal transduction pathways in a cell. (Taken from wikipedia.)

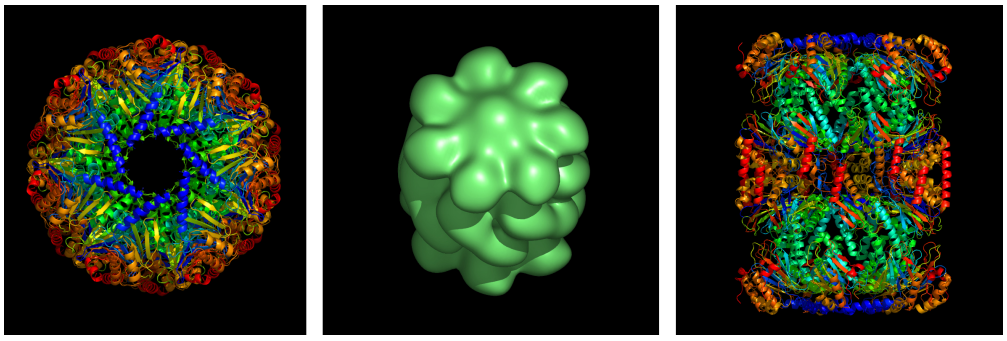

Figure 2.2: The image of a proteasome as revealed by x-ray crystallography in Mycobacterium tuberculosis. Inside the structure there is a large chamber where the denatured proteins are cleaved. (Taken from Brookhaven National Lab site.)

latter two are significant as they both directly affect the cost of screening and the ability to generate genome-scale data sets. Below we review the two main methods for large scale detection of protein interactions. Interactions data generated in the experiments are available in multiple online databases such as DIP [43], MIPS [27], BIOGRID [39] and others.

2.2.1 Yeast Two-Hybrid (Y2H)

Yeast Two-Hybrid is a technology for screening protein-protein interactions. Two-hybrid screens allow testing thousands of interactions in a single experiment. Yeast-two-hybrid works because some eukaryotic transcription factors (including, e.g., the yeast GAL4 protein) have modular structure, working through two separate activation and binding domains. A binding domain binds to a specific DNA sequence (promoter) and the activation domain recruits the transcription machinery.

To test for an interaction between proteins X and Y, two hybrid proteins should be engineered. The first one, called 'bait', is a fusion of DNA-binding domain (BD) and protein X. Notably, the hybrid can bind to DNA, but it doesn't have the activator. The second hybrid is called the 'prey' and is a fusion of DNA-activation domain (AD) and a second protein of interest Y. Hybrid proteins are injected into the yeast nucleus to perform the test.

To reliably detect the interactions, a reporter gene providing easily observable phenotype change should be chosen (e.g., color change when enabled) and integrated into the tested genome with a GAL4 promoter. A good candidate is lacZ, coding for *beta*-galactosidase which is colored blue. The Y2H is mainly used with yeast cells, but actually any organism can be used for hosting. For example, some recent studies have used E. coli for interaction testing as it was found to have several characteristics which may make it preferable to yeast, such as faster growth rate of the cells, allowing use of random libraries larger than 10^8 ([22]) and the ability to study proteins which are toxic to yeast.

In a high-throughput experiment, a pool of yeast cells transformed with different AD plasmids is mated to a chosen BD plasmid and then transferred to selective plates to select diploid cells expressing the interacting pair and activating the reporter gene. Data sets that were generated using this method for yeast include those of by Uetz et al. [15] and Ito et al. [16], reporting on thousands of

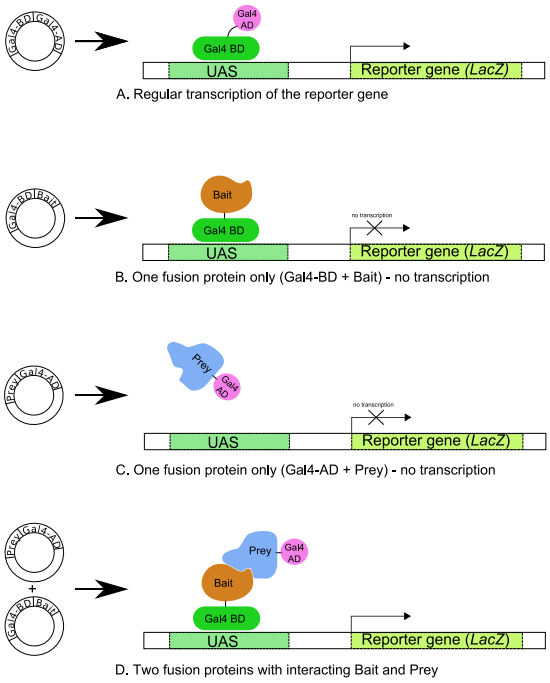

Figure 2.3: The Y2H assay scheme. GAL4 is the transcriptional gene with binding domain GAL4-BD and activation domain GAL4-AD. The reporter gene LacZ is activated when GAL4 binds to the UAS. Two hybrids are engineered, GAL4-BD with the first tested protein (bait) and GAL4-AD with the second tested protein (prey). If the bait interacts with the prey, binding and activation domains are fused close to each other and LacZ is activated. (Taken from wikipedia.)

protein-protein interactions.

2.2.2 Tandem Affinity Purification (TAP)

The TAP is a generic procedure allowing the purification of proteins expressed at their natural level in native environment. The method involves the fusion of the TAP tag to the target protein and the injection of the construction into the host cell, maintaining the expression of the fusion protein at its natural level. Next, the fusion protein and associated components are extracted from the cell (precipitated) by affinity selection on an IgG column. After washing, the TEV protease is added to release the bound material in natural conditions. The eluate is incubated with calmodulin-coated beads in the presence of calcium, to remove the TEV protease and traces of contaminants remaining after the first affinity selection. After washing, the bound material is released with EGTA (see Figure 2.4), and proteins contained therein can be analyzed by mass spectrometry [33].

The complete TAP purification and yeast strain construction can be performed in less than a month and the costs are low, allowing large-scale applications [33]. Recently, two genome wide screens for yeast were performed using this method by Krogan et al. [13] and Gavin et al. [12].

2.2.3 Limitations of Protein-Protein Interaction Detection Methods

A challenge in processing PPI networks is dealing with relatively high level of noise that are typical to current PPI data (see Section 2.2). E.g., interaction data collected using Y2H, typically contains about 50% false interactions [8, 32, 42]. To tackle this problem various techniques for evaluating reliability of PPI interactions were developed. A common approach is based on a logistic regression model as suggested by Bader et al [3] and Sharan et al. [36]. Under this model, the probability of a true interaction between two proteins is calculated using a logistic function of several random variables, reflecting the types of experiments in which the interaction was observed, and the number of times it was detected in each experiment.

The main criticism on Y2H concerns the high error rate in interaction detec-

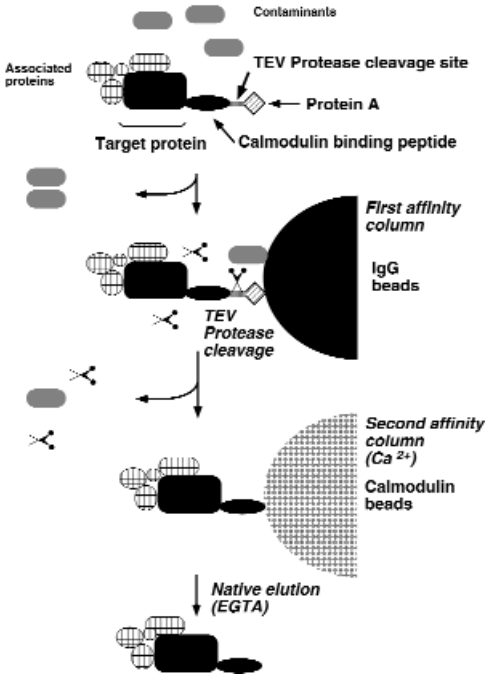

Figure 2.4: Overview of the TAP procedure. (1) The TAP tag is fused to the target protein and the construction is placed into a hosting cell, where interaction proteins bind to the target protein. (2) Tagged proteins are precipitated to IgC beads along with the interacting proteins, washed and the tags are removed by using TEV protease. (3) In a second wash stage, the TEV protease and other contaminants are removed. (4) Interacting proteins are released with EGTA. (Taken from [33].)

tion (see e.g. [8, 42]). A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae experiments [15] estimates the false positive rate in the data obtained by Y2H as 47%. In principle, the error rate can be reduced by repeating the experiment enough times, as accidental interactions are unlikely to be observed in repeated experiments. False negative rate is even higher: according to Reguly et. al. [32] the expected number of PPI interactions in the Yeast network is about 30K, while current Y2H method detects about 20K interactions. Given the estimated 47% false positives, we have almost 66% missing interactions (false negatives). Another problem with Y2H screening is that it is unable to distinguish between physiologically important protein interactions and spurious interactions, reflecting nonspecific binding of interacting domains outside of their regular cellular context.

In contrast to Y2H, in TAP experiments interactions are tested in natural conditions, which greatly improves the confidence of the detected interactions. A disadvantage of this method is its requirement of two stages of washing, making the detection of transient interactions rather hard.

It was recently claimed by Collins et al. [6] that the consolidated dataset which they have created from TAP experimental data produced by Krogan et al. [13] and Gavin et al. [12] can match the reliability of small scale experiments. Additionally, from the degree of overlap between the data sets the authors are certain that the data covers at least 80% of the interactions accessible to the TAP approach under the tested conditions.

Chapter 3

Previous Work and Our Approach

In this chapter we present the challenge of finding conserved protein complexes and review previous approaches for solving the problem and their caveats. We end the chapter with a summary of our research contribution.

3.1 The network alignment problem

Given a set of protein-protein interactions networks of different species and protein sequence similarity data for every cross-species protein pair, we wish to find subnetworks that are conserved across the species both in terms of proteins (similar sequence) and interactions (similar topology). The biological motivation for this problem comes from an assumption that conserved subnetworks are more likely to represent true protein complexes.

Formally, let graphs $G_i = (V_i, E_i)$, $1 \leq i \leq k$ denote the PPI networks of species $1 \ldots k$, where V_i is the set of proteins of species i and E_i is the set of protein-protein interactions. Let E_H be a set of edges connecting vertices in $\cup_i V_i$ as follows: for any pair of proteins $(u, v) \in V_i \times V_j, i \neq j$, there is an edge $(u, v) \in E_H$ if u and v are sequence similar (BLAST E-value$(v_m, v_n) < 1E-7$). Let $G_H = (\cup_i V_i, E_H)$ denote the graph spanned by these edges.

Assuming that conservation implies sequence similarity, a set of potentially orthologous proteins is represented by a connected subgraph of size k in G_H and includes a vertex from each species. We call such a subgraph a *k-spine*. Formally,

define a *d-subnet* as a collection U of k multi-sets $U_i = \{u_i[1], \ldots, u_i[d]\}$, each corresponding to a species $s(U_i)$, with the following properties:

- For all $1 \leq i \leq k$ and $1 \leq j \leq d$, $u_i[j] \in V_i$.
- For all $1 \leq j \leq d$, the set $U[j] = \{u_1[j], u_2[j], \ldots, u_k[j]\}$ forms a k-spine.

Finally, let $\mathscr{S}(U)$ be a scoring function over conserved subnetworks U.

Under this notation, the alignment problem can be formulated as follows: given input PPI networks of k species, sequence similarity data and a scoring function $\mathscr{S}(U)$, the output should be a collection of all conserved subnetworks U whose score exceeds a given threshold.

Unfortunately, this problem has no general-case polynomial-time algorithm. For example, it was shown by Shamir et al. [34] that the problem of searching for heavy induced subgraphs in a graph is NP-hard even when considering a single species where all edge weights are 1 or -1. This poses a serious computational barrier, making the development of a scalable solution a challenging task.

3.2 Previous Methods for Aligning PPI Networks

A variety of methods have been proposed for PPI network alignment. Below we will describe in detail three methods that have been used for aligning multiple networks: NetworkBLAST, Graemlin and CAPPI.

3.2.1 NetworkBLAST

The algorithm was developed by Sharan et al. [35]. It allows searching for linear pathways and complexes that are conserved across multiple species, currently supporting the processing of up to 3 species. The method uses a network alignment graph data structure for storing PPI data, a probabilistic log-likelihood model for alignment scoring and a greedy-search based algorithm for identifying putative protein complexes.

Network construction: As mentioned in Chapter 2, the interactions are detected using a variety of methods, providing different levels of reliability. Hence,

in a processing step interactions are assigned reliability scores using a logistic regression model.

The model defines the probability of a true interaction between two proteins as a function of the number of different experiments in which the interaction was observed and their types. Formally, given n different experiments, let $X_{uv} = (X^1_{uv} \dots X^n_{uv})$, where X^i_{uv} is the number of observations of an interaction between u and v in experiment i. Using the logistic distribution function, the probability of a true interaction T_{uv} given the observation X_{uv}, is:

$$P(T_{uv}|X_{uv}) = \frac{1}{1 + e^{-\beta_0 - \sum_{i=1}^{n} \beta_i X^i_{uv}}}$$

Applying the logistic regression method involves learning the distribution parameters β_0 ... β_n to maximize the likelihood of some training data. Since no training data are readily available, the following approach is used as proposed in [3]: Given a PPI network, an observed interaction between proteins p_1, p_2 is considered a positive example if removing it from the network leaves p_1 and p_2 at distance ≤ 2. It is considered a negative example if removing it leaves p_1 and p_2 at distance ≥ 4.

Building a network alignment graph: To facilitate searching for conserved structures across two or more networks, they are merged into a single *network alignment graph*. In this graph nodes represent sets of similar (potentially orthologous) proteins from different species and edges represent conserved protein-protein interactions.

Defining the graph for two species more formally, let V_1 and V_2 be the vertex sets (proteins) of two PPI networks. For each pair $(u, v) \in V_1 \times V_2$ of proteins with sufficient sequence similarity (BLAST E-value $< 1e - 7$), a node $\langle u, v \rangle$ is added to the alignment graph G. Nodes $\langle u_1, v_1 \rangle$ and $\langle u_2, v_2 \rangle$ are connected by an edge if both u_1, u_2 and v_1, v_2 interact in the respective PPI networks.

The definition can be easily extended to more than two species (see Figure 3.1), but practical application of this scheme to more than 3 species is too costly, due to the exponentially growing size of the alignment graph.

Scoring model: The heart of NetworkBLAST is a scoring method, based on a probabilistic model for protein complexes. This approach attempts to address

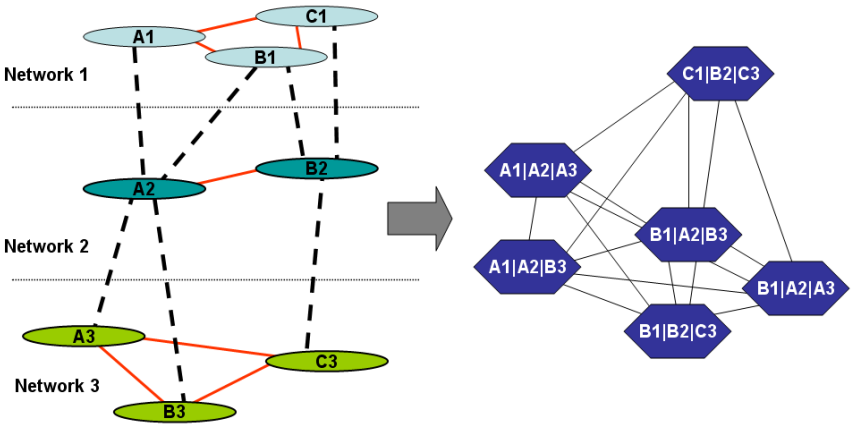

Figure 3.1: An illustration of an alignment graph constructed from PPI networks of 3 species. PPI appear as solid edges. Dashed edges denote sequence similarity relations.

the problem of noise in the protein interaction data by incorporating reliability information to it's scoring scheme. The score reflects the fit of an observed subnetwork to a functional subnetwork model versus the chance that its interactions arise at random. The functional subnetwork model assumes that each pair of proteins in a functional subnetwork should interact, independently of all other pairs, with high probability β. According to the random model, the chance for an interaction to occur depends only on the degrees of its end points.

In a single species, for a subnetwork with vertex set U, the likelihood ratio score factors over the vertex pairs in it:

$$\mathcal{L}(U) = \sum_{(u,v)\in U\times U} w(u,v)$$

where $w(v,v) = 0$ and for $u \neq v$:

$$w(u,v) = \log \frac{\beta Pr(O_{uv}|T_{uv}) + (1-\beta)Pr(O_{uv}|F_{uv})}{p_{uv}Pr(O_{uv}|T_{uv}) + (1-p_{uv})Pr(O_{uv}|F_{uv})}$$

Here O_{uv} denotes the set of experimental observations on the interaction status of u and v, T_{uv} denotes the event that u and v truly interact, and F_{uv} denotes the event the u and v do not interact. The computation of $Pr(O_{uv}|T_{uv})$ and $Pr(O_{uv}|F_{uv})$ is based on the reliability assigned to the interaction between u and v. This definition is extended to k species by treating conserved subnetworks U_i

independently and summing up the scores across the different species:

$$\mathscr{S}(U) = \sum_{i=1}^{k} \mathcal{L}(U_i)$$

The application of the definition to nodes of an alignment graph is straightforward: The nodes are decomposed to their associated proteins in each network and the score is calculated as specified above.

Search algorithm: Using the network-alignment graph to organize comparative interaction data and the presented scoring model, the problem of identifying conserved protein complexes reduces to identification of heavy subsets of interconnected nodes in the alignment graph. To tackle the problem, a variant of greedy-search is applied: First, the graph is exhaustively searched for high-scoring seeds of $d \geq 3$ nodes around each root node. Next, the seeds are greedily expanded using local search over nodes which are at distance at most two from the root node. At each step a node is added or deleted, so what the score of the modified seed increases the most. To preserve the original seed throughout this process, seed nodes are protected from being deleted.

Filtering the results: The search algorithm yields multiple overlapping subgraphs, some of which may not be true protein complexes. To filter such false positives, a statistical filtering based strategy is applied. Shortly, the algorithm is executed on randomized instances of the networks with the same node degrees, to learn the random distribution of subnetwork scores. This information is used as a significance threshold to filter out solutions found in the search phase. In addition to the statistical filtering, the resulting set of putatively conserved protein complexes must be filtered to remove overlapping solutions. The filtering is performed by applying a greedy strategy: each time the subnetwork with the highest score is selected and all subnetworks that overlap it by a certain amount are discarded.

Summary: The NetworkBLAST method is based on an alignment graph and is currently able to process of up to 3 networks in several hours. Its computational bottleneck is processing large PPI networks of multiple species, an inherent

problem for algorithms based on building a complete network alignment graph.

3.2.2 Graemlin

Another method for multiple network alignment was developed by Flanick et al. [17]. The method, called Graemlin, aligns the input networks progressively, processing a pair of most evolutionary-similar networks in each round.

Network construction: Networks are constructed by an aggregation method used was originally developed by Srinivasan et al. [38]. Similarly to the aggregation approach used in NetworkBLAST study [35], the problem is formulated as classifying interacting proteins from non-interacting pairs. A vector of attributes is the input to the binary classifier, which returns the integrated probability that the two proteins are interacting.

The attributes used in this study are: Pearson correlation between expression profiles (co-expression), Pearson correlation between phylogenetic profiles (co-inheritance), Pearson correlation between distance matrices taken element-wise (co-evolution), and average chromosomal distance between ORFs (co-location). The classifier function is trained on a set of known interactions, derived from KEGG annotations. Applying this classifier to predict interaction probabilities for all protein pairs in a genome yields a probabilistic protein interaction network. For computational efficiency a probability cut-off is used (0.25 in this study).

Pairwise network alignment: To find high-scoring alignments between a pair of PPI networks, Graemlin first creates a set of seeds. The seeds are alignments of two *d-clusters*, which are sets of d densely connected proteins. The d-clusters are calculated around each node in a PPI network by finding the $d-1$ nearest neighbors of that node in terms of interaction probabilities and may overlap each other. Then, the d-clusters are examined in pairs by a greedy algorithm to find high-scoring d-cluster pairs with a score larger than a selected threshold T.

Given the seeds, Graemlin tries to expand each into a high-scoring alignment in a manner similar to that used by other methods (e.g., NetworkBLAST [35] and MaWISh [26]). The seed extension is greedy, at each step all proteins adjacent to some protein in the alignment are added to a frontier containing candidates to be added (or removed) to the alignment. Iteratively, Graemlin selects from the

frontier a pair of proteins (one from each species) that yields the maximal increase in score when added or removed from the alignment. The extension ends when no pair of proteins from the frontier can increase the score of the alignment. To avoid an exponential increase in the size of the frontier as more nodes are added to the alignment, Graemlin uses a heuristic for selecting which nodes to add.

Scoring function (pairwise alignment): The score of an alignment has two components: the equivalence class component and the interaction component as is described next.

The *equivalence class* represents a set of proteins descending from a common ancestor. It is scored by reconstructing the most parsimonious ancestral history of the class, based on evolutionary events including sequence mutations, insertions, deletions and protein duplications. Each of these events is assigned a different probability, and the overall score is calculated as a log likelihood ratio between the probabilities assigned to the evolutionary events by an alignment and a random model.

The edges between the equivalence classes are scored using an edge scoring matrix (ESM). In the ESM raws and columns are labeled, and each cell in the ESM contains a probability distribution over edge weights. To score an alignment, each equivalence class is assigned a label and the interaction scores are calculated for each pair of equivalence classes by using a corresponding coefficient in the matrix. The ESM is used to control the topology of the conserved subnetworks, making one topology preferred over another. For instance, assigning high coefficients to diagonal entries of the matrix allows awarding paths topologies, while assignment of equal coefficients to the entire matrix awards dense topologies.

Certainly, given more than one label in the ESM, there are many possibilities to assign them to equivalence classes to score an alignment. In this case, the score is defined to be the highest which can be obtained by a labeling. To avoid exhaustive search over all possibilities, Graemlin uses a heuristic to find a good labeling.

Multiple alignment: To handle the alignment of multiple networks, Graemlin uses a progressive alignment technique. Using a phylogenetic tree, a pair of closest networks is selected and aligned, constructing product networks from the alignments as well as a list of unaligned nodes. The networks are placed at

the parent (in the tree) of the pair of networks what were aligned. While the constructed networks contain nodes that are no longer proteins but equivalence classes, the pairwise alignment and scoring methodologies remain the same. The process stops when the root of the phylogenetic tree is reached.

Notably, in parallel to each alignment network Graemlin also maintains two additional networks composed of unaligned nodes from the two original networks to enable comparisons of unaligned parts of a network, to more distant species as algorithm traverses the phylogenetic tree.

Filtering results: Similarly to other algorithms based on seed expansion, the constructed alignments have overlaps. To resolve the problem, Graemlin traverses the alignments and makes them disjoint by removing overlapping equivalence classes from all but one of the alignments. This is done by choosing alignments and equivalence classes resulting in minimal decrease of score of either alignment.

Summary: The approach was shown to be very scalable. In particular, the authors presented results for aligning of 10 microbial networks. A clear drawback of a progressive alignment approach is loss in sensitivity, as after each pairwise alignment step the algorithm has to decide which information should be discarded, to prevent exponential growth in the processed data. Since the decision is taken before considering data in networks which are added to the alignment at later stages, weak signals may be lost. Another weakness of the method is the requirement to assign network proteins to distinct equivalence classes early. The clustering is performed greedily, prior to considering information from networks of other species, which may lead to suboptimal solutions.

3.2.3 CAPPI

CAPPI is a recent method for multiple network alignment developed by Dutkowski et al. [9]. The method is based on reconstructing a putative common-ancestor PPI network of the aligned species.

Clustering proteins to putatively orthologous families: The first step of the reconstruction involves clustering all input proteins to distinct families. The proteins in each cluster are assumed to descend from a single common ancestral

protein. Later, each such cluster will form a node in the reconstructed ancestral PPI network. The families are found by running the TRIBE-MCL algorithm developed by Enright et al. [10], using BLAST E-values as pairwise distances between the proteins.

Reconstructing the evolutionary history in each cluster: Families of putatively orthologous proteins are processed in an attempt to reconstruct the evolutionary history of their member proteins. In each family the protein sequences are aligned using CLUSTALW [41]. Then, the protein distance matrix is calculated using the PROTDIST procedure and a phylogenetic tree is reconstructed using the neighbor joining algorithm from Phylip package [21]. Finally, the gene tree is reconciled with the species tree of the aligned organisms to infer duplication and speciation events using the NOTUNG method of Durand et al. [7].

Reconstructing the ancestral network: Let $G_{i,j} = (V_{i,j}, E_{i,j})$ be the undirected graph representing the protein network of species i after the j-th evolutionary event (duplication or speciation) occurring in this species. In this notation $G_{1,0}$ is the ancestral graph, having exactly one protein for each protein family (cluster), corresponding to the common ancestor protein of all genes in the family. At this stage the method tries to determine the probability of interaction between nodes in $G_{1,0}$ based on the observed PPIs of the input species and the sequences of evolutionary events for the corresponding proteins.

The algorithm starts from $G_{1,0}$ and builds all the networks $G_{i,j}$ using the following procedure. First, $G_{i,j-1}$ is copied to $G_{i,j}$. By definition, the difference between $G_{i,j-1}$ and $G_{i,j}$ is a result of a duplication (d_j) or speciation (s_j) event. In case of a duplication event, a node n_p in $G_{i,j-1}$ is duplicated, creating two nodes n_a and n_b in $G_{i,j}$. Next, each edge (n_p, n_x) in $G_{i,j-1}$ is translated to edges (n_a, n_x) and (n_b, n_x) in $G_{i,j}$ with probability p_D. For each node n_y in $E_{i,j-1}$ which is not connected to n_p, we add edges (n_a, n_y) and (n_b, n_y) independently with probability $\delta_D/|V_{i,j}|$. In case of speciation event, a protein node is copied between the species defining a set of similar rules, creating graphs $G_{LS(i),0}$ and $G_{RS(i),0}$ from $G_{i,j}$, where $LS(i)$ and $RS(i)$ denote the left and right sons species of the speciation event.

In this model the probability of interaction between proteins in $G_{i,j}$ is determined by the sequence of duplication and speciation events leading from the ancestral network $G_{1,0}$ to $G_{i,j}$, and the interactions in the ancestral network. Viewing the ancestral graph edges as random variables and formulating the problem of finding the ancestral topology ($G_{1,0}$) from a sequence of graphs $G_{i,j}$ as a Bayesian network inference problem, allows solving it efficiently with a simple version of the message-passing algorithm (see e.g., [29]).

Identifying conserved ancestral modules: The ancestral network $G_{1,0}$ is a complete graph, where edges are assigned a score corresponding to the probability of interaction between the connected proteins. To decompose the ancestral network to modules, the algorithm chooses a threshold value t and eliminates from the graph all edges scored below the threshold, decomposing the giant component $G_{1,0}$ to multiple, heavily interconnected components. The cut-off value t is chosen such, that remaining interaction in the ancestral graph are statistically significant compared to the background model where probability of nodes to interact is only dependent on nodes degrees. The significance is estimated by running the algorithm on random networks obtained by shuffling edges of input networks, preserving degrees of original nodes (a similar technique was applied in [25, 26, 35]).

Summary: The approach should scale well for multiple networks due to efficiency of the algorithm used for the initial clustering of networks proteins to homologous families. At the same time, the authors have only provided results for the alignment of 3 species networks and we are not aware of any attempts to use the algorithm for more species. A disadvantage of this method is its dependency on the quality of the early clustering of proteins to distinct homologous families.

3.3 Contribution of this research

Searching for protein complexes is an important problem in molecular biology. With a constantly growing pool of PPI data, the scalability of multiple network alignment algorithms becomes increasingly important. In the previous sections

we described several existing methods for multiple network alignment and pointed the caveats of each approach. Methods based on the construction of alignment graph, like NetworkBLAST [35], have limited scalability. Progressive alignment based methods like Graemlin [17] scale well but suffer from inherent sensitivity problems. CAPPI [9] addresses the complexity problem by early clustering of the input proteins to non-overlapping families but is sensitive to the quality of the clustering and its application to more than 3 network are not yet available.

In this research we focused on the development of an algorithm that allows scalable processing of multiple PPI networks without the shortcomings of previous methods. The algorithm achieves a significant improvement in terms of running time and memory consumption (compared to its predecessor NetworkBLAST), retaining the accuracy of an exhaustive search based approach. The heart of our approach is a representation of multiple PPI networks in a layered data structure that is linear in their sizes. We show how to efficiently find putative orthologous sets of proteins in this layered graph, avoiding the construction of a complete alignment graph. Our method allows the alignment of 10 networks with tens of thousands of protein-protein interactions each in minutes.

Another tool which we developed as part of this research is a web-server implementing the NetworkBLAST algorithm [35]. The web-server allows the detection of putative protein complexes in a single species network or subnetworks conserved across PPI networks of two different species. It is able to extract protein-protein interactions from raw PSI MI 2.5 XML format (as available in, e.g., DIP [43]), assign interaction reliability scores automatically, and provide a graphical representation of the detected subnetworks. This makes it an easy-to-use and friendly tool for identifying protein complexes in protein-protein interaction networks.

Chapter 4

The Algorithm

In this chapter we present a novel algorithm for the multiple network alignment problem. We analyze the complexity of the algorithm and discuss possible problem restrictions and computational trade-offs. Finally, we provide details on the implementation and choice of parameters.

4.1 Data representation

Consider the definition of the network alignment problem given in Section 3.1. We represent the input data using a k-layer graph, which we call a *layered alignment graph*. Each layer corresponds to a species and contains the corresponding network. Additional edges connect proteins from different layers if they are sequence similar. Formally, layer i has a set V_i of vertices and a set E_i of edges. Additionally, we have a set of *inter-layer* edges denoted by E_H. Let $G_H = (\cup_i V_i, E_H)$ denote the graph restricted to the inter-layer edges. Let n be the maximum number of proteins in a species. Let m be the maximum number of sequence similarity edges in G_H between any pair of species.

The relation between an alignment graph (see Section 3.2.1) and a layered alignment graph should be clear: while in the former every set of potentially orthologous proteins is represented by a vertex; in the latter such a set is represented by a subgraph of size k (a k-spine) which includes a vertex from each of the layers. Key to the algorithmic approach presented below is the assumption that a k-spine corresponding to a set of truly orthologous proteins must be connected and, hence, admits a spanning tree. Thus, we can identify all potential vertex

sets inducing k-spines by looking for trees instead.

A collection of (connected) k-spines induces a candidate conserved subnetwork. We score it using a likelihood ratio score developed by Sharan et al. [35] (see Section 3.2.1). We denote by $\mathscr{S}(U)$ the score of a subnetwork with vertex set U.

4.2 The search algorithm

The main algorithmic task is to look for high scoring d-subnets, for a fixed d. This problem is computationally hard even when there is only a single network, and edge-weights are restricted to $+1$ for all edges, and -1 for all non-edges [34]. Thus, we resort to a greedy heuristic which starts from high weight seeds and expands them using local search. Such greedy heuristics have been successfully applied to search for conserved subnetworks in a network alignment graph [17, 26, 35].

There are two sub-tasks we need to tackle: (i) computing high weight seeds; and (ii) extending a seed. We provide algorithmic solutions for both tasks below.

4.2.1 Computing seeds

We start by computing d-subnets as *seeds*, where d is a small constant. Notably, even when $d = 2$, we do not know of any algorithm better than the naive approach, which involves looking at all pairs of k-spines. This $O(n^{dk})$ time algorithm is intractable for typical networks, so we consider two alternative assumptions on the inter-layer edges that reduce the computational complexity while retaining the sensitivity of the algorithm.

The first assumption asserts that the k-spines of a seed support the same topology of inter-connections. This is motivated by the observation that proteins within the same pathway or complex are typically present or absent in the genome as a group [30]. Thus, we consider the following problem:

Problem 1: ***d-identical-spine*-subnet :** *Compute a set of d k-spines with identical topologies and maximum score.*

Theorem 1: The *d-identical-spine*-subnet problem admits an $O(k^3 m^d 3^k)$ solution.

Proof: First, consider the case where each of the d k-spines is restricted to be a path (Figure 4.1). This implies that the d-subnet itself can be considered as a path $U_1, U_{2,} \ldots, U_k$. For a subset of species S, let $\mathscr{S}(U, S)$ denote the score of the best d-subnet that uses only species in S, and consists of a path that ends with U. To compute $\mathscr{S}(U, S)$, note that we only need to recurse using the predecessor of U in the path. Formally:

$$\mathscr{S}(U,S) = \begin{cases} \max\limits_{\substack{(U,W)\in E_H \\ s(W)\in S\setminus\{s(U)\}}} \mathscr{S}(W, S\setminus\{s(U)\}) + \mathcal{L}(U) & \text{if } |S|>1 \\ \mathcal{L}(U) & \text{if } |S|=1 \end{cases}$$

The recursion is computed for every pair $(U, W) \in E_H$ ($O(k^2 m^d)$ in total), and each subset S (2^k possibilities). For a proposed multi-set W, it takes $O(k)$ time to check whether $s(W) \in S$. Thus, the overall time is $O(k^3 m^d 2^k)$.

A similar recursion can be applied when searching for k-spines that are trees with identical topology. For a subset of species S, let $\mathscr{S}(U, S)$ denote the score of the best d-subnet that uses only the species in S, and consists of a tree rooted at U. Then for $|S| > 1$:

$$\mathscr{S}(U,S) = \max_{\substack{(U,W)\in E_H, S_1\subset S \\ s(U)\in S_1, s(W)\in S\setminus S_1}} \mathscr{S}(U,S_1) + \mathscr{S}(W, S\setminus S_1)$$

To analyze the complexity, notice that there are $O(3^k)$ variations of $(S, S_1, S \setminus S_1)$. Thus, the overall time is $O(k^3 m^d 3^k)$. ♣

A second, slightly different assumption is based on the phylogeny (described as a rooted, binary tree T) of the investigated species. Consider a set of nodes a, b, c whose underlying species form the phylogenetic triple $(s(a), (s(b), s(c)))$. We assume that if a, b, c are connected via inter-layer edges, then b and c must be connected. This implies that we can restrict our attention to k-spines that are *guided* by the phylogeny T in the following sense: any restriction of the k-spine to species that form a clade in T is a subtree of the k-spine. Note that two guided spines can have very different topologies (see Figure 4.2).

Problem 2: **The *d*-*guided-spine*-subnet problem:** *Compute a set of d*

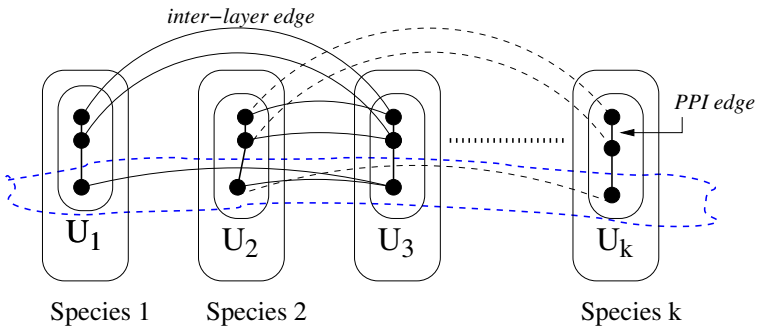

Figure 4.1: A seed defined by a d-identical-spine subnet, where the k-spines are restricted to be paths with identical topology. The dashed line encloses one of the three k-spines.

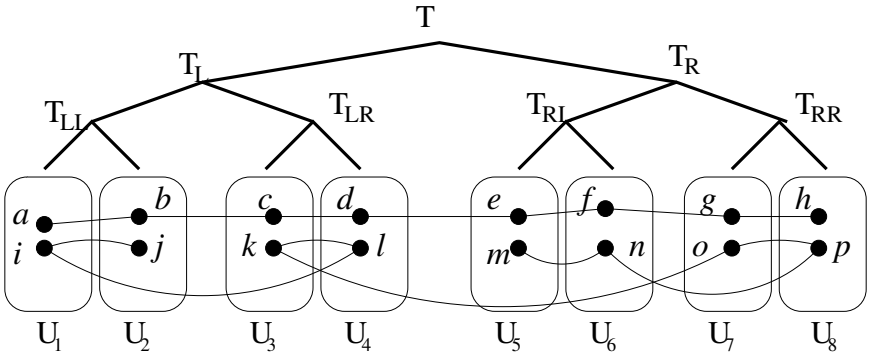

Figure 4.2: Sketch of a 2-*guided-spine*-subnet . Note that while the paths of the two k-spines have different topologies, they are both guided by the underlying tree. Following the notation in the proof of Theorem 2, let $U = \{a, j\}, W = \{h, m\}$, and consider two possible distant sets $X = \{d, k\}$ and $Y = \{e, o\}$. By definition, $T_{LL}(U \cup X) = \{a, j\}, T_{LR}(U \cup X) = \{d, k\}, T_{RL}(Y \cup W) = \{e, m\}, T_{RR}(Y \cup W) = \{h, o\}$. Hence, $\mathscr{S}(U, W, T) \geq \mathscr{S}(\{a, j\}, \{d, k\}, T_L) + \mathscr{S}(\{e, m\}, \{h, o\}, T_R) \geq \mathscr{S}(\{a, i\}, \{b, j\}, T_{LL}) + \mathscr{S}(\{c, k\}, \{d, l\}, T_{LR}) + \mathscr{S}(\{e, m\}, \{f, n\}, T_{RL}) + \mathscr{S}(\{g, p\}, \{h, o\}, T_{RR}) \geq \mathcal{L}(a, i) + \mathcal{L}(b, j) + \mathcal{L}(c, k) + \mathcal{L}(d, l) + \mathcal{L}(e, m) + \mathcal{L}(f, n) + \mathcal{L}(g, o) + \mathcal{L}(h, p)$.

k-spines, guided by the underlying phylogeny, with maximum score.

Unfortunately, we do not know of any efficient algorithm better than the naive $O(n^{kd})$ for this problem. However, we show a better solution when the k-spines are restricted to be paths guided by the phylogeny.

Theorem 2: The d-guided-spine-subnet problem can be solved in $O(k(kn)^{2d}|E_H|^d)$ time when restricted to paths.

Proof: Consider a subtree T of the phylogeny with subtrees T_L, T_R, respectively. Clearly, each of the d paths will have one end-point in T_L, and the other in T_R. However, the order of the species along these paths is not identical. Therefore, we work with size d subsets U which are not restricted to be within a single species, but instead can span any species in T.

Let $\mathscr{S}(U, W, T)$ denote the best score of a set of d paths guided by a subtree T of the phylogeny, such that $s(U) \subseteq T_L, s(W) \subseteq T_R$ are the end nodes. At the base of the recursion T consists of a single node and $\mathscr{S}(U, U, T) = \mathcal{L}(U)$.

Denote the root of T by $root(T)$. For a node u, define its *distant set*:

$$\mathcal{D}_T(u) = \{x | \mathrm{LCA}_T(s(\{u\}), s(\{x\})) = \mathrm{root}(T)\}$$

where $LCA_T(a, b)$ is the least common ancestor of a and b in T. Extend this to d elements by defining $\mathcal{D}_T(U) = \{X |\ \mathrm{LCA}_T(s(\{u[j]\}), s(\{x[j]\})) = \mathrm{root}(T)\ \forall j\}$. By definition, if $X \in \mathcal{D}_T(U)$ then for all j: $s(x[j]) \in T_L, s(u[j]) \in T_R$ or $s(x[j]) \in T_R, s(u[j]) \in T_L$. Define $T_L(X)$ ($T_R(X)$) as the set of all vertices in X with species in T_L (T_R) respectively. Finally, let T_{LL}, T_{LR} be the two subtree of T_L and let T_{RL}, T_{RR} be the two subtrees of T_R. Then, as exemplified in Figure 4.2:

$$\mathscr{S}(U, W, T) = \max_{\substack{X \in D_{T_L}(U) \\ Y \in D_{T_R}(W) \\ (X,Y) \in E_H}} (\mathscr{S}(T_{LL}(U \cup X), T_{LR}(U \cup X), T_L) + \mathscr{S}(T_{RL}(Y \cup W), T_{RR}(Y \cup W), T_R))$$

Intuitively, the recursion works by dividing the problem of finding a d-guided-subnet (U, W) with respect to a phylogenetic tree T, into two sub-problems on the two sub-trees, T_L and T_R. For the running time, note that there are

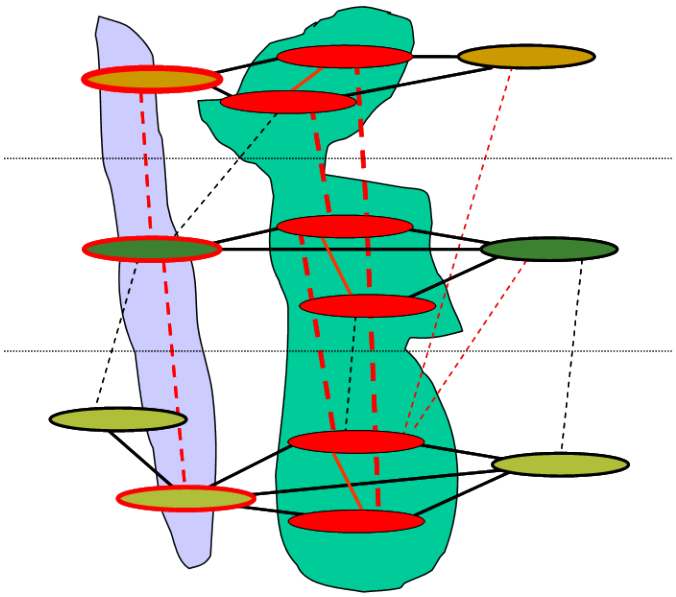

Figure 4.3: Illustration of the seed extension phase of the algorithm. The figure shows a layered data-structure for 3 species, a 2-spine seed denoted by a green area in the middle, and an extension candidate spine (blue area to the seed left) interconnected with the seed by PPI edges (solid black lines).

$O((kn)^{2d}|E_H|^d)$ possible choices for (U, X, Y, W). Since a choice of U and W implies T, the overall time is $O(k(kn)^{2d}|E_H|^d)$. ♣

4.2.2 Extending a seed

The next phase of the algorithm is performing an iterative expansion of the seed by adding, in each iteration, the k-spine that contributes the most to the score (see Figure 4.3). Let us denote by H the current seed, and by $\mathscr{S}(v, S)$ the score of the best partial extension of H by a subtree that is rooted at vertex v and visits the species in S. Then $\mathscr{S}(v, S)$ can be computed using the following recursive relation:

$$\mathscr{S}(v,S) = \begin{cases} \max\limits_{\substack{(v,w)\in E_H, S_1 \subset S \\ s(\{v\})\in S_1, s(\{w\})\in S\setminus S_1}} \mathscr{S}(v,S_1) + \mathscr{S}(w, S\setminus S_1) & \text{if } |S|>1 \\ \mathcal{L}(v) & \text{if } |S|=1 \end{cases}$$

The overall complexity is $O(k|E_H|3^k)$.

There are two speedups one can introduce to this basic extension scheme. The first is to set in advance the order of the species along the phylogenetic tree, eliminating the 3^k factor. We term this variant *restricted order* as opposed to the previous *relaxed order* variant. The second is to constrain k-spines to paths (rather than trees), obtaining an $O(k|E_H|2^k)$ time algorithm.

4.3 Implementation notes

We have designed a software package, *NetworkBLAST-M*, implementing the multiple network alignment approach outlined above. The implementation allows looking for 2-identical-spine seeds with relaxed and restricted orders. For efficiency reasons, we restricted the seed vertices in each network to be of distance at most 2 from one another.

The final collection of conserved subnetworks was filtered to remove redundant solutions. This was done using an iterative greedy procedure that selects each time the highest scoring subgraph and removes all subgraphs intersecting it by more than 50%. For two conserved subnetworks A and B, containing $|A|$ and $|B|$ proteins, respectively, the intersection level is computed as the number of common proteins over $\min\{|A|, |B|\}$.

To verify that using 2-identical-spines are adequate for our problem, we analyzed alignment nodes within conserved network regions output by NetworkBLAST [35] for different network sets. When aligning the networks of yeast, worm and fly (NetworkBLAST data set), in 85% of the cases the pertaining alignment nodes respected the yeast-worm-fly phylogeny-based ordering. In two additional microbial network sets (C. jejuni, E. coli, H. pylori and C. crescentus, V. cholerae and H. pylori; Graemlin data set) more than 95% of the alignment nodes respected the respective phylogeny-based orientation. Moreover, 72% of the alignment nodes actually formed cliques in G_H.

We have also experimented with the two seed extension variants presented here. Our results in this regard indicate that the relaxed-order variant yields higher sensitivity on certain data sets while showing similar specificity (see Table 5.6). In the restricted order variant, the best sensitivity is achieved with spines whose orientation respects the phylogenetic tree. Interestingly, the difference in results between different orientations of spines in restricted order is not

as significant as could be expected, probably because a large fraction of spines in functionally enriched alignments form cliques in G_H and such spines are robust to ordering restrictions.

Chapter 5

Experimental Results

5.1 Data Description and Parameters

We applied our algorithm to eukaryotic and microbial PPI networks, summarized in Table 5.1. The three eukaryotic networks were taken from [35] (Network-BLAST data set) or [9] (CAPPI data set), and the microbial networks were taken from [17] (Graemlin data set). As in [35], we used a BLAST E-value threshold of 10^{-7} for sequence similarity, ensuring a corrected significance value of 0.01.

5.2 Validation Techniques

We evaluated the identified conserved subnetworks by computing the functional coherency of their member proteins with respect to the biological process annotation of the gene ontology (GO) [2], for each species separately. To this end, we used the GO TermFinder tool [5] to compute empirical enrichment p-values, and corrected for multiple testing using the false discovery rate procedure [4]. For each species we report the percent of functionally coherent subnetworks discovered, and the number of distinct GO categories they cover. The first measure quantifies the specificity of the method, and the second provides an indication of the sensitivity of the method.

Table 5.1: A summary of the PPI networks analyzed in this study.

Species (taxon id)	#Proteins	#PPIs
Graemlin data set		
S. coelicolor (100226)	6678	230409
E. coli E12 (83333)	4087	216326
M. tuberculosis (83332)	3457	128932
S. typhimurium (99287)	4239	94609
C. crescentus (190650)	3341	40524
V. cholerae (243277)	2948	36038
S. pneumoniae (170187)	1843	25726
C. jejuni (192222)	1442	22116
H. pylori (85962)	1070	12943
Synechocystis sp. (1148)	2371	69439
NetworkBLAST data set		
S. cerevisiae (4932)	4738	15147
C. elegans (6239)	2853	4472
D. melanogaster (7227)	7165	23484
CAPPI data set		
S. cerevisiae (4932)	4726	15103
C. elegans (6239)	2619	3950
D. melanogaster (7227)	7032	20782

5.3 Comparison to Extant Methods

To establish the validity of our method, we first compared it to NetworkBLAST [35]. NetworkBLAST is an exhaustive approach that relies on explicitly constructing a network alignment graph and, hence, is limited in application to the alignment of up to 3 networks. Both methods use the same scoring function and scoring parameters were set equal for both methods for fair comparison. The results in Table 5.2 show that the performance of NetworkBLAST-M is comparable to that of NetworkBLAST. The latter has higher specificity, but fewer GO categories enriched. The sensitivity of NetworkBLAST-M further improves when using the relaxed-order variant. Notably, the application of NetworkBLAST-M took less than 30 seconds in both configurations, while NetworkBLAST's run took more than six hours.

To compare to the CAPPI method [9], we applied both methods to the data set which was used in the original publication, executing them with default pa-

rameters. In order to apply NetworkBLAST-M to these data we assigned identical scores (0.8) to all interactions, as no reliability information was available for them. As shown in Table 5.3, the NetworkBLAST-M yields generally higher specificity and consistently higher sensitivity than CAPPI.

Table 5.2: A comparison of NetworkBLAST-M and NetworkBLAST on three eukaryotic networks from the NetworkBLAST data set. For these networks NetworkBLAST produced 59 conserved regions, while NetworkBLAST-M identified 64 regions in the restricted-order variant and 92 in the relaxed-order variant.

Species	Specificity (%)	# GO categories enriched
NetworkBLAST		
S. cerevisiae	100.0	14
C. elegans	88.0	13
D. melanogaster	94.9	16
NetworkBLAST-M restricted order		
S. cerevisiae	100.0	29
C. elegans	65.6	29
D. melanogaster	98.4	37
NetworkBLAST-M relaxed order		
S. cerevisiae	94.6	45
C. elegans	67.0	29
D. melanogaster	90.1	41

Next, we compared the performance of NetworkBLAST-M to that of Graemlin [17] on a set of 10 microbial networks. Graemlin's results were taken from the original publication, considering only alignments which contain all 10 species (a total of 21 conserved regions). NetworkBLAST-M was applied only in the restricted-order variant due to the high computation burden. The algorithm detected a total of 33 conserved network regions. As summarized in Table 5.4, NetworkBLAST-M outperforms Graemlin, providing uniformly higher specificity and sensitivity.

Statistics on the running times of NetworkBLAST-M on different sets of microbial networks with 3-10 species are given in Table 5.5. Evidently, the restricted-order variant is considerably faster than the relaxed-order variant and can process up to 10 networks in minutes.

Statistics on the running times of NetworkBLAST-M on different sets of

Table 5.3: A comparison of NetworkBLAST-M and CAPPI on three eukaryotic networks from the CAPPI data set. For these networks CAPPI reported 38 conserved regions and NetworkBLAST-M identified 74 regions in the restricted-order variant and 98 in the relaxed-order variant.

Species	Specificity (%)	# GO categories enriched
CAPPI		
S. cerevisiae	91.4	29
C. elegans	70.0	21
D. melanogaster	72.7	20
NetworkBLAST-M restricted order		
S. cerevisiae	97.3	40
C. elegans	59.2	25
D. melanogaster	85.1	39
NetworkBLAST-M relaxed order		
S. cerevisiae	99.0	46
C. elegans	61.1	21
D. melanogaster	86.7	48

microbial networks with 3-10 species are given in Table 5.5. As evident, the restricted-order variant is considerably faster and can process up to 10 networks in minutes.

Table 5.4: A comparison of NetworkBLAST-M and Graemlin on 10 microbial networks. Results are provided for nine of the ten species for which we had gene ontology information (for Synechocystis we did not have functional information readily available).

Species	Specificity (%)	# GO categories enriched
NetworkBLAST-M restricted order		
S. coelicolor	100	17
E. coli E12	90	16
M. tuberculosis	87.9	17
S. typhimurium	93.1	14
C. crescentus	84.8	15
V. cholerae	90.6	16
S. pneumoniae	97.0	14
C. jejuni	96.2	12
H. pylori	92.3	13
Synechocystis	N/A	N/A
Graemlin		
S. coelicolor	71.4	12
E. coli E12	76.5	10
M. tuberculosis	76.9	8
S. typhimurium	81.3	10
C. crescentus	86.7	11
V. cholerae	80.0	9
S. pneumoniae	71.4	8
C. jejuni	76.9	9
H. pylori	56.3	8
Synechocystis	N/A	N/A

Table 5.5: NetworkBLAST-M run-time as a function of the number of species and the size of the layered alignment graph. All the tests were performed on Intel Xeon 3.06GHz 3GB memory machine.

#Species	#Nodes	#PPI edges	#Sequence similarity edges	Restricted order run time (sec)	Relaxed order run time (sec)
3	8132	102288	26834	40	44
5	11945	193843	57142	72	1587
7	17236	301365	103887	83	46686
10	31458	877032	327219	140	N/A

Table 5.6: NetworkBLAST-M performance evaluation of relaxed vs. restricted order configurations for all possible orientations of spines in a yeast-worm-fly data set.

Species	Specificity (%)	# GO categories enriched
Relaxed order		
S. cerevisiae	94.6	45
C. elegans	67.0	29
D. melanogaster	90.1	41
(Y,(W,(F))) order		
S. cerevisiae	100.0	32
C. elegans	65.6	29
D. melanogaster	98.4	37
(Y,(F,(W))) order		
S. cerevisiae	98.5	35
C. elegans	61.9	23
D. melanogaster	92.4	33
(W,(Y,(F))) order		
S. cerevisiae	98.0	27
C. elegans	62.0	17
D. melanogaster	88.0	29

Chapter 6

The NetworkBLAST Web Server

In parallel to working on *NetworkBLAST-M* we have developed a web-server implementing the original NetworkBLAST algorithm (see Section 3.2.1 for its description). Shortly, the algorithm constructs an alignment of the analyzed networks, which is searched for conserved protein complexes. Each candidate complex is scored by its fit to a protein complex model, which assumes a certain density of interactions within complexes, versus the likelihood that it arises at random. The server currently supports the analysis of one or two networks and the output consists of a set of putative protein complexes along with their visualization. In contrast to previous tools for online protein network comparison, which support only single queries of a given pathway/complex in a network of interest, the NetworkBLAST web-server allows the uncovering of all conserved complexes across two networks.

6.1 Input

NetworkBLAST can operate in two-species or single-species modes. For two species, the input consists of the two respective PPI network files and a sequence similarity file containing BLASTP E-values between pairs of proteins from each of the species. In the single-species mode, only PPI data for one species are needed.

A PPI file may be in text or XML format. In the former case, each row represents an interaction and contains the IDs of the interacting proteins pair and a confidence value for it. Alternatively, the user can upload a PSI MI 2.5 file,

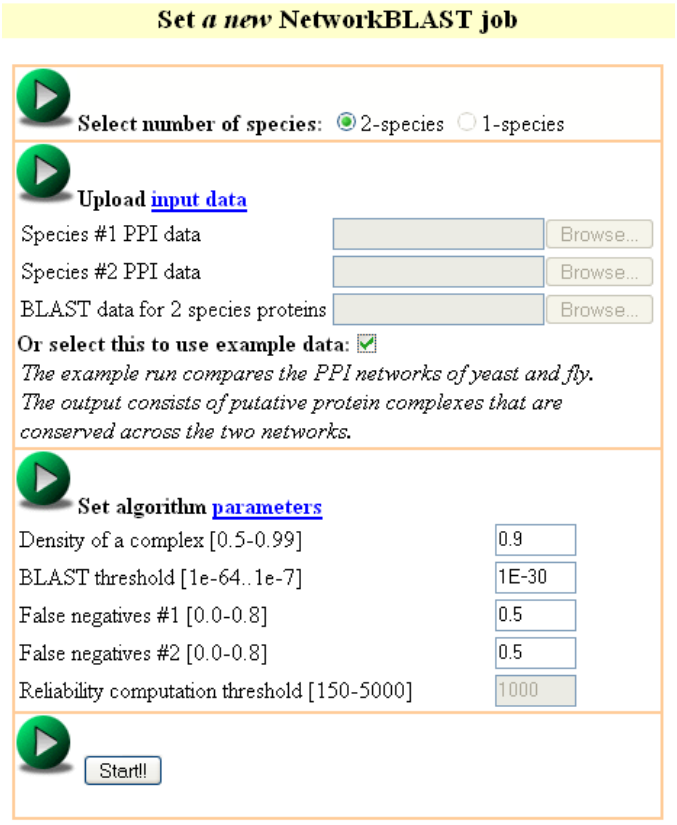

Figure 6.1: The web-server's main page, exemplifying the setting of a new network alignment job for sample PPI networks.

as e.g. available in the Database of Interacting Proteins (DIP) [43]. In this case, the server uses information on the types of experiments in which each interaction was detected to automatically infer a confidence value for it; the computation is based on the logistic regression scheme of Sharan et al. [35]. The sequence similarity file is a text file, in which each row contains a pair of proteins from different species and their BLASTP E-value.

The user can control several parameters of the algorithm, including the sequence similarity threshold for potential orthology, the way experiments are categorized for interaction reliability computation (see web-site) the density of the sought protein complexes, and the estimated rate of false negatives in each network. The latter two parameters affect the scoring of the candidate complexes (See Section 3.2.1).

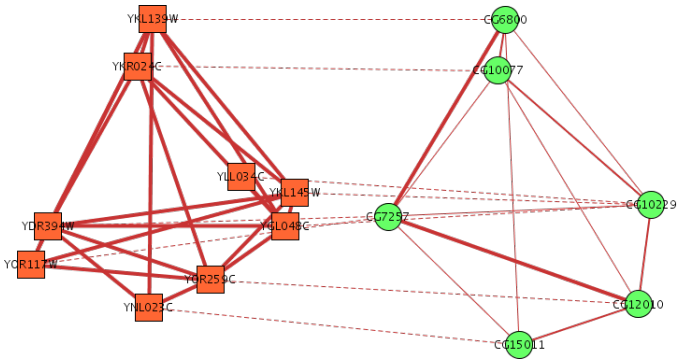

Figure 6.2: A representative yeast-fly conserved complex from NetworkBLAST's output. Yeast proteins appear in orange; fly proteins appear in green. Sequence-similar proteins are connected with dashed lines. Solid lines represent PPIs, with the line width corresponding to the reliability of the corresponding interaction.

6.2 Output

The web-server generates an html page with links to images of the discovered complexes, as well as textual graph data files that can be viewed using the Cytoscape software [14]. The output putative complexes are sorted by their score. Their images are generated automatically using the DualLayout plugin of the Cytoscape software. For a single species, the images show the member proteins of each putative complex with edges connecting interacting proteins and edge width corresponding to the interaction's reliability. In two-species mode, the images show aligned putative complexes across two species (drawn side by side), where dashed lines connect orthologous proteins and solid lines denote PPIs (Figure 6.2).

6.3 Application

NetworkBLAST's performance depends on the sizes of the input networks and the similarity between their proteins. One of the main factors influencing the running time is the size of the constructed network alignment graph. This in turn depends on the sequence similarity threshold required for two proteins to be considered potentially orthologs. For efficiency reasons, the server runs are currently limited to alignment graphs with up to 5,000 nodes. Larger graphs can be handled using an offline version of the program, available for download at the

website.

6.4 An Example Run

We demonstrate the utility of the algorithm by presenting an example run on the PPI networks of S. Cerevisiae and D. Melanogaster. These networks are also available to the user as default inputs. Protein-protein interaction data for yeast and fly were downloaded from DIP [43] and contained 15,147 interactions among 4,738 proteins in yeast and 23,484 interactions among 7,165 proteins in fly. To assign confidence scores to these interactions we used the logistic- regression-based scheme employed in [35]. The false negative rates were estimated at 50% [35]. The BLAST threshold was set to 1E-30 to allow fast running time. The analysis revealed 49 putative conserved complexes. We tested for functional coherency of these complexes using the GoTermFinder tool [5]. 78% of the yeast complexes and 63% of the fly complexes were functionally enriched (after correcting for multiple testing using the false discovery rate procedure), serving as a validation of these predictions.

Chapter 7

Conclusions

We have provided two tools for multiple network alignment. The first tool is a web-server implementing the NetworkBLAST algorithm [35], which allows analyzing a single species or aligning PPI networks of two species to search for putative protein complexes. PPI network data are uploaded in tab-separated or PSI-MI XML formats [31], and the server provides a graphical visualization of the discovered protein clusters.

The second tool is an algorithm for efficiently aligning multiple PPI networks. The algorithm uses a novel representation of multiple protein-protein interaction networks as a layered graph, showing a significant improvement in speed and memory usage compared to alignment-graph based algorithms. It is shown to outperform previous approaches based on progressive alignment and common ancestor network reconstruction ideas.

Future research includes a more extensive comparison of the different seed computation variants presented here, as well as experimenting with other scoring functions. For example, it could be interesting to incorporate into the NetworkBLAST-M framework a scoring function which is based on modeling protein complex evolution, extending the method of [18] to multiple networks. Finally, NetworkBLAST-M should be implemented within our web-server, which will greatly expand its applicability.

The development of network alignment techniques and tools, such as the two described here, is crucial to the study of protein network evolution and is expected to become increasingly important as protein-protein interaction databases continue to grow in size and species coverage.

Bibliography

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

[2] M. Ashburner et al. The gene ontology consortium. gene ontology: Tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

[3] J.S. Bader, A. Chaudhuri, J.M. Rothberg, and J. Chant. Gaining confidence in high-throughput protein interaction networks. *Nature biotechnology*, 22:78–85, 2004.

[4] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*, 57 (1):289–300, 1995.

[5] 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*, 20:3710–15, 2004.

[6] S.R. Collinsa, P. Kemmerena, X.C. Zhaog, J.F. Greenblatth, F. Spencerg, F.C.P. Holstegee, J.S. Weissmana, and N.J. Krogan. Toward a comprehensive atlas of the physical interactome of saccharomyces cerevisiae. *Mol. Cell. Proteomics*, 6(3):439–450, 2007.

[7] B. Vernot D. Durand, B. V. Halldorsson. A hybrid micro-macroevolutionary approach to gene tree reconstruction. *Journal of Computational Biology*, 13(2):320–335, 2006.

[8] C. M. Deane, L. Salwinski, I. Xenarios, and D. Eisenberg. Protein interactions: Two methods for assessment of the reliability of high-throughput observations. *Mol. Cell. Proteomics*, 1:349–356, 2002.

[9] J. Dutkowski and J. Tiuryn. Identification of functional modules from conserved ancestral protein-protein interactions. *Bioinformatics*, 23:149–158, 2007.

[10] A.J. Enright, S. Van Dongen, and C.A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res*, 30:1575–84, 2002.

[11] A.C. Gavin et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415:141–147, 2002.

[12] A.C. Gavin et al. Proteome survey reveals modularity of the yeast cell machinery. *Nature*, 440:631–636, 2006.

[13] N.J. Krogan et al. Global landscape of protein complexes in the yeast saccharomyces cerevisiae. *Nature*, 440:637–43, 2006.

[14] P. Shannon et al. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504, 2003.

[15] P. Uetz et al. A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, 403:623–627, 2000.

[16] T. Ito et al. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci. USA*, 98:4569–4574, 2001.

[17] J. Flannick et al. Graemlin: general and robust alignment of multiple large interaction networks. *Genome Research*, 16:1169–81, 2006.

[18] E. Hirsh and R. Sharan. Identification of conserved protein complexes based on a model of protein network evolution. *Bioinformatics (ECCB'06)*, In press, 2006.

[19] Y. Ho et al. Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature*, 415:180–183, 2002.

[20] T. Ito, T. Chiba, and M. Yoshida. Exploring the yeast protein interactome using comprehensive two-hybrid projects. *Trends Biotechnology*, 19:23–27, 2001.

[21] Felsenstein J. Phylip - phylogeny inference package. *Cladistics*, 5:164–6, 1989.

[22] J. K. Joung, E.I. Ramm, and C.O. Pabo. A bacterial two-hybrid selection system for studying protein-dna and protein-protein interactions. *PNAS*, 97:7382–7387, 2000.

[23] M. Kalaev, V. Bafna, and R. Sharan. Fast and accurate alignment of multiple protein networks. *RECOMB*, 2008.

[24] M. Kalaev, T. Ideker M. Smoot, and R. Sharan. Networkblast: comparative analysis of protein networks. *Bioinformatics*, 24(4):594–596, 2008.

[25] B.P. Kelley et al. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proc. Natl. Acad. Sci.*, 100:11394–9, 2003.

[26] M. Koyuturk et al. Pairwise local alignment of protein interaction networks guided by models of evolution. *Journal of Computational Biology*, 13:182–99, 2006.

[27] H.W. Mewes, J. Hani, F. Pfeiffer, and D. Frishman. MIPS: a database for protein sequences and complete genomes. *Nucleic Acids Research*, 26:33–37, 1998.

[28] NetworkBLAST web-server. http://www.cs.tau.ac.il/ roded/networkblast.htm.

[29] R.E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2003.

[30] M. Pellegrini, E.M. Marcotte, M.J. Thompson, D.Eisenberg, and T.O. Yeates. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *PNAS*, 96:4285–4288, 1999.

[31] Molecular interaction XML. http://www.psidev.info/index.php?q=node/60.

[32] T. Reguly et al. Comprehensive curation and analysis of global interaction networks in saccharomyces cerevisiae. *J Biol.*, 5:11, 2006.

[33] G. Rigaut, A. Shevchenko, B. Rutz, M. Wilm, M. Mann1, and B. Sraphin. A generic protein purification method for protein complex characterization and proteome exploration. *Nature Biotechnology*, 17:1030–1032, 1999.

[34] R. Shamir, R. Sharan, and D. Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144:173–182, 2004.

[35] R. Sharan et al. Conserved patterns of protein interaction in multiple species. *Proc. Natl. Acad. Sci.*, 102:1974–9, 2005.

[36] R. Sharan and T. Ideker. Modeling cellular machinery through biological network comparison. *Nature Biotechnology*, 24:427–33, 2006.

[37] R. Sharan, T. Ideker, B.P. Kelley, R. Shamir, and R.M. Karp. Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. *Journal of Computational Biology*, 12:835–846, 2005.

[38] B.S. Srinivasan, A. Novak, J. Flannick, S. Batzoglou, and H. McAdams. Integrated protein interaction networks for 11 microbes. *Proceedings of the Tenth Annual International Conference on Computational Molecular Biology*, page pp. 1.14, 2006.

[39] C. Stark et al. BioGRID: a general repository for interaction datasets. *Nucleic Acids Research*, 34:D535–D539, 2006.

[40] U. Stelzl et al. A human protein-protein interaction network: a resource for annotating the proteome. *Cell*, 122:830–2, 2005.

[41] J.D. Thompson, D.G. Higgins, and T.J. Gibson. Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22:4673–4680, 1994.

[42] C. von Mering et al. Genome evolution reveals biochemical networks and functional modules. *Proc Natl Acad Sci.*, 100:15428–33, 2003.

[43] I. Xenarios et al. DIP, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res.*, 30:303–5, 2002.