Evolution of protein-protein interaction networks

Janusz Dutkowski

Faculty of Mathematics, Informatics and Mechanics

University of Warsaw

A thesis submitted for the degree of

Doctor of Philosophy

April 2009

Advisor: prof. dr hab. Jerzy Tiuryn

Author's declaration:
Aware of legal responsibility I hereby declare that I have written this dissertation myself and all the contents of the dissertation have been obtained by legal means.

April 15, 2009 ....

Date *Janusz Dutkowski*

Supervisor's declaration:
The dissertation is ready to be reviewed

April 15, 2009 ....

Date *prof. dr hab. Jerzy Tiuryn*

Abstract

Large-scale protein-protein interaction (PPI) networks are now available for human and many model organisms. The arising challenge is to analyze these data to reveal the basic components and organization of the cellular machinery. Pioneering studies have shown that cross-species comparison is an effective approach for uncovering key modules in PPI networks. Early successes have in turn stimulated the research for new methods, with a more solid grounding in mathematical models, and better scalability, to allow multiple network comparison.

We develop a novel framework for comparing PPI networks across species, providing new insights into the evolution of these systems. Our approach is based on the reconstruction of a hypothetical PPI network of the common ancestor of the considered species. The reconstruction algorithm is built upon a proposed model of protein network evolution, which takes into account phylogenetic history of the proteins and the rewiring of their interactions. Initial application of our procedure to networks of *D. melanogaster*, *C. elegans* and *S. cerevisiae* reveals that the most probable ancestral interactions often correspond to known protein complexes. Further, we extend our phylogeny-based framework to provide a method for transferring and integrating PPI evidence from multiple datasets and species. The method is used to predict unknown protein associations and provide interaction-level confidence scores for seven eukaryotic networks, including the human interactome. We also develop an EM-based procedure for estimating the parameters of our model from data and apply it to derive rates of conserving and neutral PPI evolution. Based on the evolutionary rates, we construct a network of the most conserved co-functioning protein families.

Keywords: protein-protein interactions, biological networks, network evolution, network alignment, Bayesian networks, inference, message passing, expectation maximization.

ACM Classification: J.3 Biology and genetics.

Acknowledgements

First, I want to thank my advisor, Jerzy Tiuryn, for working with me on this project, giving me guidance and the freedom to pursue my interests, and setting a great example of how to approach research problems.

I am also grateful to all the members of the computational biology group at our department, especially to Ania Gambin for drawing me into my first research project, Boguś Kluge who has always been a great colleague and Bartek Wilczyński for being the best possible lab mate and for sharing his infinite technical insight, whenever I asked. Special thanks also goes to Norbert Dojer, Przemek Biecek and Paweł Górecki for very helpful feedback on sections of the manuscript. I also want to thank Prof. Andrzej Jerzmanowski and Maciej Kotliński from the Institute of Biochemistry and Biophysics (IBB PAS) for discussions on the results concerning *A. thaliana*.

Many thanks to Anna Osmańska-Zych for advice and guidance on the formal regulations of the PhD studies. I agree with Filip Murlak that she is one of the most helpful people in the department.

Over the last three years, I have greatly benefited from visits to numerous institutions. I would like to thank Prof. Lengauer at the MPI in Saarbrücken, Prof. Waterman, Dr. Chen and Prof. Sun at USC in Los Angeles, Dr. Godzik at the Burnham Institute in San Diego, and Prof. Gross at the KITP at UC Santa Barbara for making these visits possible.

Most importantly, I'm grateful to my parents and Maja. I thank my Mom and Dad for their support and very encouraging yet permissive directions they have always given me. I also want to thank my Dad specifically for reading a draft version of this thesis and suggesting corrections which improved its clarity. Maja, thanks for being there for me and for your patience and understanding throughout the writing process – I'm almost done.

During the preparation of this thesis I was supported by the Polish Ministry of Science grants No 3 T11F 021 28, PBZ-MNiI-2/1/2005 and 1589/B/P01/2008/35. Computational resources were provided by the CoE BioExploratorium project WKP 1/1.4.3/1/2004/44/44/115.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the completion of the Human Genome Project (Venter *et al.*, 2001; International Human Genome Sequencing Consortium, 2001) researchers have gained a comprehensive catalogue of genes coding for functional molecules in our cells. These molecules, including primarily proteins, but also non-translated RNA sequences, constitute a set of parts available in each cell under certain conditions. Numerous studies have shown that these components almost never work alone. It has become evident that it is not just the set of parts, but rather the complex system of interactions between them which determines specific cellular functions and ultimately gives rise to life. The cellular system is controlled on many levels, each governed by different types of molecular interactions. For example, transcriptional regulation, managed by a network of inter-dependencies between genes, regulatory regions, promoters and suppressors, allows some genes to be expressed while keeping others silent – depending on the type of the cell, the stage in the cell cycle and other conditions. The availability of gene products is one of the factors predetermining possible physical and biochemical interactions, which themselves form interconnected system-level networks.

Molecular systems biology is an area of research which focuses on assembling and analyzing biological networks as models of cell machinery. It is a rapidly growing multidisciplinary field which benefits from developments in biology, physics, mathematics and computer science. Systems approach is often considered a paradigm alternative to the classical reductionist approach. While the latter has been successful in identifying components and many individual interactions, it fails to provide means to understand how these components relate with each other and function cooperatively in biological systems. Systems biology tries to address these problems through measuring multiple components simultaneously and by integrating data with mathematical models (Ideker *et al.*, 2001; Sauer *et al.*, 2007).

Proteins play a key role in biological systems. They participate in practically every process taking place in the cell. They carry out essential tasks in cell cycle, signalling and immune response. They act as enzymes catalyzing biochemical reactions during metabolism. Some proteins also have mechanical functions (e.g. in muscle contraction) and structural roles (e.g. in the cytoskeleton). Proteins carry out their functions through physical interactions with other proteins in complexes and pathways. These interactions are diverse and abundant. Some of them are remarkably robust while others are transient, occurring occasionally, under specific conditions. The number of interaction partners varies greatly among proteins. While most have only a few interactions, there are some which have even hundreds. Different complexes and pathways are inter-connected forming larger networks involved in achieving complicated tasks. Protein-protein interactions (PPIs) are of great interest due to their fundamental role in cellular processes, as well as their significance for theoretical studies on the emergence of complex systems. PPI networks constitute invaluable models for interrogating developmental and disease mechanisms, as well as elucidating individual protein function (Cusick *et al.*, 2005). During the last decade they have been established as one of the primary levels at which cellular systems can be studied.

1.1 Large-scale PPI networks

The investigations of protein-protein interactions at systems level have been induced by the development of high-throughput experimental systems. Since the pioneering studies by Uetz *et al.* (2000) and Ito *et al.* (2001), which provided the first large-scale snapshots of the interactome of *S. cerevisiae* (baker's yeast), the yeast-two-hybrid (Y2H) system has been applied to map PPIs in many model organisms, as well as in human (Rual *et al.*, 2005; Stelzl *et al.*, 2005). Other techniques, such as tandem affinity purification coupled with mass spectrometry

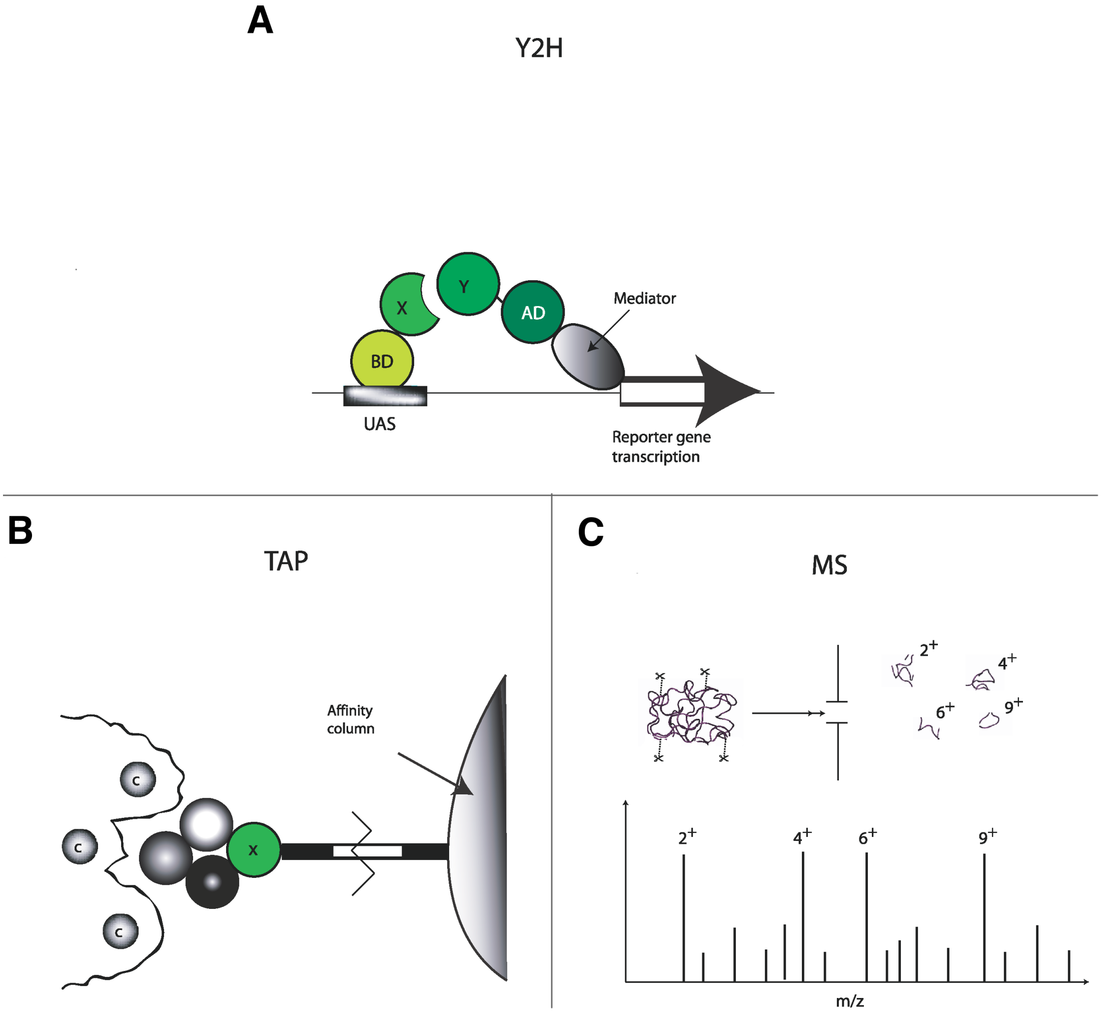


Figure 1.1: *Experimental techniques for identifying protein-protein interactions. The Y2H system (A) is based on a two-domain transcription factor Gal4 which activates the expression of a reporter gene His3. The transcription factor is split and the DNA-binding domain (BD) is fused to protein X, while the activation domain (AD) is fused to protein Y. If X and Y interact then His3 is expressed and the cells can grow under histidine-deficient conditions. Tandem affinity purification (B) is used to purify protein complexes. A protein of interest X is fused with two affinity tags (black boxes) separated by a protease cleavage site (white box). After washing through two affinity columns, the remaining binding partners of protein X can be detected by mass spectrometry (C) which converts protein peptide molecules into ions and identifies them by measuring their mass-to-charge ratios. Figure adapted from Shoemaker & Panchenko (2007b).*

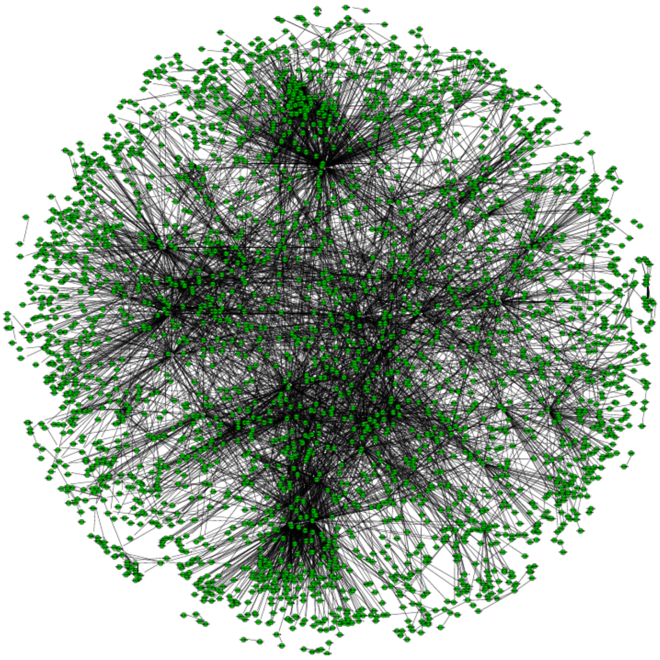


Figure 1.2: *Experimentally derived PPI network of baker's yeast (S. cerevisiae). The nodes of the graph represent proteins and edges represent physical protein-protein interactions. Network diagrams in this thesis were prepared using the Cytoscape software platform (Shannon* et al.*, 2003).*

(TAP-MS) (Gavin *et al.*, 2006; Krogan *et al.*, 2006), have been successful in identifying co-complex associations (i.e. proteins co-occurring within the same complex). We provide a short overview of Y2H and TAP-MS in Fig. 1.1. For more details on these and other experimental methods see Shoemaker & Panchenko (2007b) and Uetz *et al.* (2008). We will assume that the output of a single experiment can be represented in the form of an undirected graph corresponding to a PPI network (see Fig. 1.2).

Unfortunately, the available experimental techniques are far from perfect, both in terms of their accuracy, as well as coverage. For instance, the yeast interactome has recently been estimated to contain from around 37000 up to even 75500 protein interactions between approximately 6000 proteins. Although already over 80000 yeast PPIs have been reported, given the estimated false positive rates of the experiments, the yeast interactome is suggested to be roughly 50% complete

(Hart *et al.*, 2006). Using a more conservative definition and omitting indirect co-complex associations, Yu *et al.* (2008) estimate the number of yeast interactions to be ∼18000 and conclude that three idependent Y2H assays cover only around 20% of this amount. In case of human, the entire interacome is estimated to contain from approximately 130000 (Venkatesan *et al.*, 2009) up to 650000 (Stumpf *et al.*, 2008) PPIs and is covered in roughly 10% (Hart *et al.*, 2006; Venkatesan *et al.*, 2009).

Since the publications of the first large-scale studies, many doubts and criticism have been expressed in the literature, regarding the low overlap between independent screens. The Y2H screens of Uetz *et al.* (2000) and Ito *et al.* (2001) identified 1519 and 4549 PPIs, respectively, with an overlap of less than 20% of the smaller set. Initial investigations have pointed to a high false positive rate of the Y2H system (Mrowka *et al.*, 2001; von Mering *et al.*, 2002; Bader & Hogue, 2002). More recent studies (e.g. Yu *et al.*, 2008) suggest that the low overlap can largely be explained by low sampling sensitivity (low coverage) and differences in assay types. Most authors agree that all metioned factors contribute to the observed situation to some extent. Even a very small false positive rate might dramatically impact the result, due to the dominating number of non-interacting pairs. Given the average current estimates, the expected number of true yeast interactions is approximately 0.2% of all possible protein pairs (∼18 million). This implies that an experiment with a false positive rate of 1% would identify about 5 false interactions for every true one. If all pairs were tested, the result would contain ∼179640 false positives. Factoring in accuracy and sampling sensitivity of available methods, it would require tens of proteome-scale screens to identify each mappable interaction with high confidence (Schwartz *et al.*, 2008). Some transient PPIs might not be recovered by any presently available technique. In summary, none of the existing methods can provide a complete and error-proof interaction map of an organism, within reasonable time and respecting budget limitations. Therefore it is crucial to develop systematic approaches that integrate evidence from complementary experiments and provide quantitative confidence scores for each possible interaction.

A separate problem, also calling for computational approaches, concerns the analysis of complex PPI networks and organising them into models of cellular machinery (see Panchenko & Przytycka, 2008, for a recent review). *A priori* there is little knowledge of the role of individual network components and their assignment into complexes and pathways. In certain situations it is possible to map onto the network previously established information on individual proteins and interactions, adding a new perspective to the available data. This type of analysis has proven very useful in associating selected subnetworks with various phenotypic traits including diseases (for a review see Ideker & Sharan, 2008). In the vast majority of cases, however, the role of individual network components and their ensembles is unknown.

One of the ways of identifying functional network components is *via* cross-species analysis. As in the case of biological sequence data, the comparative approach demonstrates high potential. The basic motivation behind it is the principle of evolutionary conservation of functional units of the cell. Just like individual genes and protein sequences are retained in many species, the common patterns of protein interactions, fulfilling essential functions, are expectedly retained as well. Cross-species comparison of PPI networks enables functional annotation of proteins, prediction and verification of protein interactions and, ultimately, identification of essential cellular units.

In this thesis we concentrate on two problems important to the understanding of protein-protein interaction networks. The first one is related to the identification of subnetworks (or network modules) performing distinct functions. The second is the overall improvement of PPI networks through the prediction of missing protein interactions and identification of false positives. We consider both of these problems from the evolutionary angle and address them using systematic comparative analysis. In the following sections we review relevant computational frameworks proposed previously and describe the basic motivations behind our approach.

1.2 Network alignment: identifying conserved subnetworks

One of the key approaches to the analysis of large PPI networks is through network alignment, or comparison of two or more networks to uncover similar regions (see Sharan & Ideker, 2006; Yosef *et al.*, 2008, for reviews). Intuitively, we are interested in identifying subnetworks (network modules) which are conserved in the compared species (see Fig. 1.3). The conservation is assessed both on the level of nodes (respective protein sequences should be similar) and on the level of edges (corresponding nodes should have similar patterns of interactions). The precise formulation of alignment depends on the particular approach.

In a pioneering study Kelley *et al.* (2003) construct an alignment graph in which nodes represent pairs of homologous proteins (one from each of the two compared networks) and edges represent conserved interactions. In practice, homology is established based on sequence similarity. The authors use a randomized dynamic programming algorithm to search through the alignment graph for conserved interaction pathways (simple paths). This approach is extended in Sharan *et al.* (2005a) to enable the search for conserved complexes. A new scoring function is developed and high-scoring dense subgraphs of the alignment graph are identified by greedily expanding small subgraph seeds. A similar procedure, but with a different evolutionary motivated scoring of nodes and edges, is presented by Koyuturk *et al.* (2006). Sharan *et al.* (2005b) propose a modified version of their previous method which aligns three (rather than two) networks simultaneously.

A drawback of the alignment graph is that it includes a node for every tuple of similar proteins (one from each input network). The commonly used similarity functions (e.g. based on BLAST E-value) generally impose a many-to-many correspondence between proteins, which causes the size of the alignment graph to grow exponentially with the number of aligned networks. Flannick *et al.* (2006) present a different approach to network alignment, which addresses this problem by greedily assigning the aligned proteins to non-overlapping homology classes and progressively aligning multiple input networks. The algorithm also allows searching for a wider range of subnetworks (not limited to pathways or dense clusters) defined by the user.

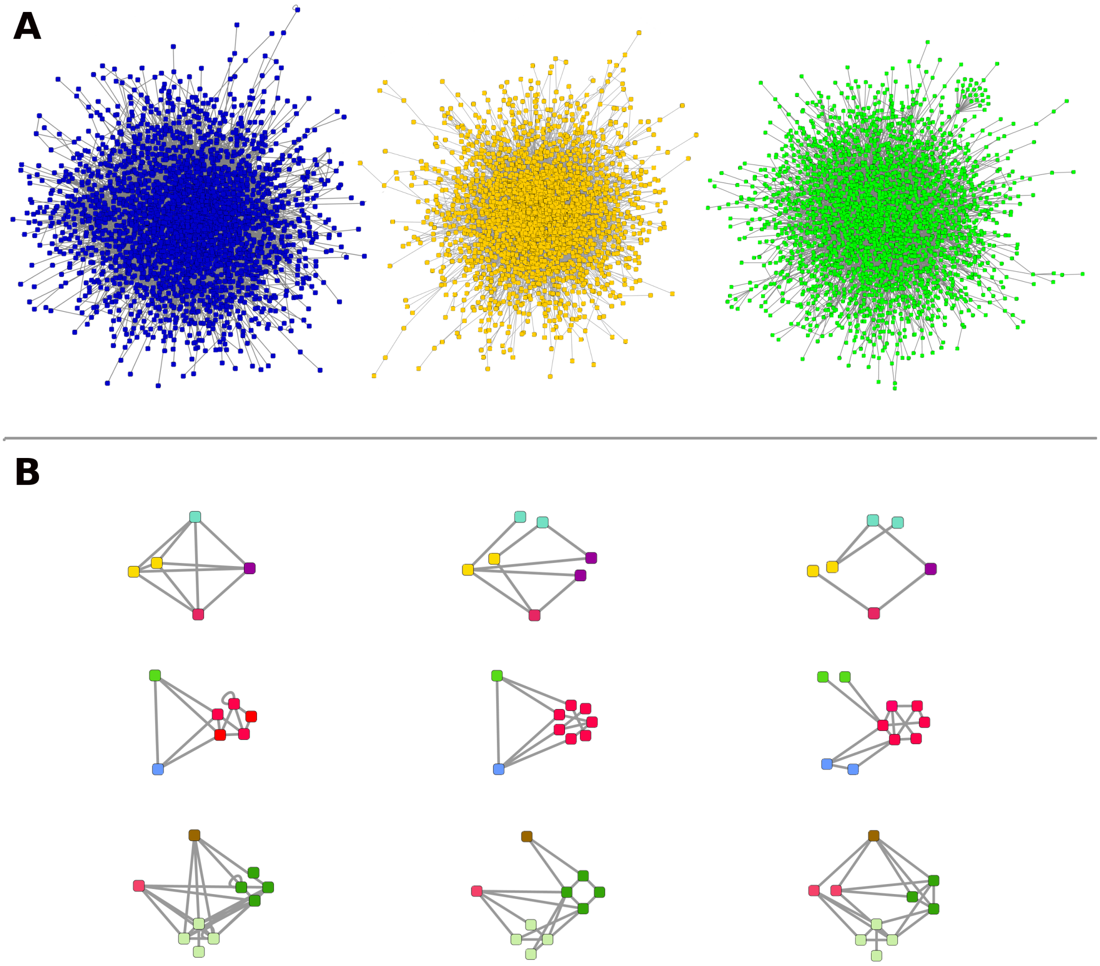


Figure 1.3: *A conceptual illustration of conserved network modules identified by network alignment. PPI networks of three different species (blue, yellow and green) are shown in (A). Three examples of connected subnetworks conserved in the considered interactomes are shown in (B). Each row in (B) presents an instance of one network module for each of the three species. Node colors in (B) denote homology relationships (sequence similarity) and establish correspondence between nodes. A protein may have many homologs in its own species, as well as in other species. In a conserved module, nodes of the same color should have a similar profile of interactions. Ideally, an interaction between nodes of two given colors should be always present or always absent (the latter situation is usually not interesting). In practice, perfect conservation is rare. Interactions of a given type might be absent completely in some species or carried out only by selected proteins.*

Other approaches to network alignment and modifications of the previous procedures have been devised more recently. Hirsh & Sharan (2006) update the scoring function of their aligner, assigning probabilities to interaction loss and emergence in a single evolutionary step leading from an ancestral module (protein complex) to the observed interaction patterns. Berg & Lassig (2006b) develop a new approach in which the evolutionary conservation of nodes and edges is treated on equal basis. They heuristically solve a quadratic assignment problem to identify the highest-scoring mapping between nodes of two networks. Li *et al.* (2007) phrase the network alignment problem in terms of integer quadratic programming, preselecting the aligned subgraphs and aligning them globally. Narayanan & Karp (2007) apply a top-down approach to detect maximal matching subnetworks. They iterate the *match* step which removes all proteins without counterpart in the other network, and the *split* step which partitions the networks into connected components. Owing to a local and monotonic match function, the algorithm finds the optimal solution in polynomial time. In some cases, however, it has to be coupled with a heuristic clustering procedure, in order to identify biologically meaningful results. Among the most recently developed methods is an approach by Singh *et al.* (2007), inspired by Google's PageRank algorithm (Brin & Page, 1998). It solves an eigen value problem, corresponding to random walks in the alignment graph, to recover a global mapping of nodes between two networks. Also very recently new updates of previous procedures have been proposed by Kalaev *et al.* (2008) and Flannick *et al.* (2008). The former provides a new representation of the alignment graph and restricts it only to certain types of homology relationships, enabling fast multiple network alignment. In the latter work, the authors enhance their previous solution with a more elaborate scoring function and an iterative procedure based on subgradient descent for learning the function's free parameters from data.

It is worth explaining that each of the mentioned methods formulates and solves a quite different mathematical problem. The notion of an evolutionarily conserved subnetwork is not precisely defined. The best one can do is to postulate desirable characteristics based on previously described examples and relevant biological knowledge. In general, many mathematical formulations of network alignment result in NP-hard problems, related to the subgraph isomorphism problem. Thus various types heuristics and simplifying assumptions often have to be applied. Another important issue is the evolutionary relevance of the identified alignments. As suggested by Sharan & Ideker (2006), network alignment should progress towards methods more strongly rooted in models of network evolution. This aspect was not fully explored by previous studies. Specifically, while evolutionary motivated alignment scores were considered (Koyuturk *et al.*, 2006; Flannick *et al.*, 2006; Hirsh & Sharan, 2006; Berg & Lassig, 2006b), none of the previous methods modeled the process of network growth and divergence in time, taking into account the possible states of the network at subsequent evolutionary stages. Our approach presented in Dutkowski & Tiuryn (2007) and further extended in this thesis attempts to address this goal. Instead of developing a scoring function which would capture the postulated characteristics of conserved modules, we attempt to reconstruct the evolutionary history of protein networks and, ultimately, discover ancestral network components best supported by the input data. The reconstruction process is guided by phylogenetic trees, representing the evolution of network nodes, and assumes a probabilistic model of interaction dynamics.

1.3 Computational methods for predicting protein interactions

The second type of methods considered here are computational approaches for predicting PPIs. These approaches complement experimental methods and can be used to validate noisy data and select new targets for screening experiments (Shoemaker & Panchenko, 2007a). Over the past few years many such techniques have been proposed (see Valencia & Pazos, 2002; Shoemaker & Panchenko, 2007a; Valencia & Pazos, 2008, for reviews). The available algorithms explore various types of experimental sources and apply different modeling frameworks. As an example, Enright *et al.* (1999) and Marcotte *et al.* (1999) infer PPIs from sequence data by identifying distinct proteins in the species of interest which are fused together into one protein in another species. Pellegrini *et al.* (1999) construct phylogenetic profiles based on the presence or absence of homologous proteins in many species. These profiles are clustered to determine functionally related and potentially interacting proteins. Various classification-based approaches have also been applied to predict PPIs (Bader *et al.*, 2004; Chen & Liu, 2005; Bleakley *et al.*, 2007). Heterogeneous data sources, including mRNA co-expression, co-essentiality and co-localization, have been integrated in a Bayesian framework by Jansen *et al.* (2003). Other authors explore the premise that interacting proteins often co-evolve and use similarity of phylogenetic trees as evidence of protein interactions (Pazos & Valencia, 2001; Jothi *et al.*, 2005; Juan *et al.*, 2008). Another popular framework uses maximum likelihood estimation (MLE) for inferring the probability of domain-domain interactions from PPI data (Deng *et al.*, 2002; Liu *et al.*, 2005; Lee *et al.*, 2006). Methods most relevant to our analysis, which identify highly probable PPIs through integration and transfer of interaction evidence between species, are described next.

1.3.1 Transferring interaction evidence across species

The comparative approach, useful for identifying conserved functional modules, also provides a powerful basis for inferring the most reliable interactions and transferring them to other organisms. In its simplest form, this idea is implemented in the interolog (the term interlog is also used) mapping method (Matthews *et al.*, 2001). The method predicts an interaction between a pair of proteins (a,b) if in another species there exists a known interaction between a pair (a',b'), where a' and b' are orthologs of a and b, respectively. In practice, orthology is concluded based on high sequence similarity. Integration of PPI evidence across species can also be achieved at the level of conserved domains (independently stable protein parts). Liu *et al.* (2005) devise a maximum likelihood method, similar to the one by Deng *et al.* (2002), but using data from multiple organisms. In summary, they estimate the probability of interactions between each pair of considered domains, based on the PPI evidence from multiple species. Inferred domain-domain interactions constitute integrated evidence, which is in turn used to predict protein-protein interactions. A similar method, but using heterogeneous data sources (including protein fusion and Gene Ontology annotations), is used by (Lee *et al.*, 2006). In general, combining interaction evidence from different species makes PPI predictions more robust to experimental noise. False positive observations are unlikely to be reproduced across multiple species (Sharan & Ideker, 2006). Furthermore, evolutionarily conserved interactions are expectedly biologically significant. Evolutionary pressures are more likely to constrain functional units, such as protein complexes, than single interactions (Beltrao & Serrano, 2007). Hence, if an interaction has experimental support in datasets from diverse species, it is likely to be part of a significant functional unit. Highly probable interactions identified in a subset of species can also be transferred to other species (Bork *et al.*, 2004), as was done by (Sharan *et al.*, 2005b) to predict missing interactions within conserved protein modules.

1.4 Our approach

The methods used for aligning PPI networks and methods for predicting interactions are quite diverse, both at the level of input data, as well as the applied modeling frameworks. As often the case in computational biology and other applied sciences, we are given some general characteristics of the desired output, but the ways of achieving it are left open. The precise mathematical formulation of the problem, motivated by biological considerations, is already part of the solution.

In this thesis we develop a unified approach for comparing PPI networks. It addresses the two challenges described in the previous sections: network alignment and interaction inference. We now explain the specific motivations behind our approach which made it different from methods proposed previously.

We attempted to develop a method that would be well grounded in a mathematical model of network evolution. Considerable amount of research has already been devoted to random graph models, including ones which aim to describe the evolution of PPI networks. The networks observed today were not created independently from scratch, but instead evolved from a smaller ancestral network. During the evolution, many proteins of the original network were duplicated (some of them multiple times), forming families of homologous proteins. Many of the interactions were conserved in the new proteins, while others were lost in one or both of the duplicates. New interactions could also have formed to accommodate new functions. Finally, some of the proteins might have been lost. At distinct time points spaciation events took place, enabling two copies of the network (one for each resulting species) to evolve independently from each other. We model the discrete steps of this growth process using a version of a duplication and divergence model. It allows us to account for possible changes in the protein space, as well as network rewiring events, i.e. interaction emergence and loss. In our model the rewiring events are associated with protein duplications and speciations, which are recovered from protein sequence data based on phylogenetic analysis. The reconstructed protein evolution serves as a backbone to study the evolution of protein interactions. We believe that this approach coincides well with the concepts of evolutionary systems biology as described by Medina (2005) and addresses the needs for a model-based network comparison framework postulated by Sharan & Ideker (2006).

We address the network alignment problem by reconstructing a hypothetical ancestral network, based on the protein sequences and interactions found in the input data. Nodes of this network represent ancestral proteins, descendants of which have been conserved in the observed input networks. Edges of the network are assigned weights which denote the probability of interaction between adjacent ancestral proteins. These probabilities are calculated as the posterior probabilities given the interaction data in the input networks and the evolutionary model, by which the input networks are assumed to have evolved from their common ancestor. The regions of the ancestral network with high interaction probability can therefore be regarded as a merged representation of the common proteins and interaction patterns that have been conserved in the input networks. An additional layer in our model enables the integration of PPI datasets from multiple experiments and across species. This provides us with an evolutionary-based approach for identifying the most reliable interactions and transferring them to relevant proteins in other organisms.

Additional important considerations include the ability to compare multiple networks and identify conserved subgraphs of unconstrained topology. The former is achieved through the direct use of protein families, which group the nodes of the compared networks. The number of protein families does not significantly increase with the number of species, allowing a tractable solution to the alignment problem. The inherent ability to recover subnetworks of arbitrary topology is due to a very selective search for the most conserved pairwise associations, which considers the evolution of each protein family. This selectivity results in decomposition of the network into specific, functionally related, connected components. One final advantage of the presented framework is that it provides the basis to distinguish the modes of network evolution and infer the relative weights of neutral and selective dynamics – challenges put forward by Berg & Lassig (2006a). At the end of this thesis, we describe a procedure for learning the parameters of our model from data, providing a way to infer rates of neutral and conserving PPI evolution.

1.5 How this thesis is organized

In Chapter 2 we discuss popular methods for comparing biological sequences, identifying protein families and reconstructing phylogenetic trees. In Chapter 3 we review selected developments in the theory of random graph models, which lay the foundations for characterizing and understanding the evolution of large network systems. In Chapter 4 we review the Bayesian network modeling framework, define the inference problem and describe Pearl's message passing algorithm for exact inference in polytrees. Chapters 2, 3 and 4 contain preliminary material. They are presented for completeness, in an attempt to make the discussion self-contained. The reader familiar with these topics can move straight to Chapter 5 and reference the preceding chapters if necessary.

In Chapter 5 we describe our network alignment framework and apply it to recover conserved functional modules in yeast, worm and fly. The material of this chapter was originally presented in Dutkowski & Tiuryn (2007). In Chapter 6 we address the problem of integrating and reconciling PPI datasets within and across species, in order to infer missing associations and filter out false positive ones. We extend our framework and apply it to reconcile seven eukaryotic networks, including the human interactome. In Chapter 7 we develop a procedure to learn the parameters from data using the expectation maximization (EM) scheme. We also provide preliminary experimental results on detection of conserving interaction evolution among selected pairs of protein families. Chapters 5, 6 and 7 should best be read in the presented order. Each of them successively builds on the preceding material. In Chapter 8 we summarize our results and mention possible ways of enhancing our framework. We also describe new research perspectives related to biological network analysis and discuss the possible application of our results towards the comparison of regulatory networks and networks of genetic interactions.

Chapter 2

Protein sequence alignments, families and phylogenies

In this chapter we review basic methods used in protein sequence analysis, which we will later apply to identify protein families and reconstruct their phylogenetic history. We note that this overview is by no means exhaustive, neither in the depth of provided description, nor in the collection of covered methods. For each of the problems mentioned here many different algorithms have been developed. We use protein sequence analysis only to prepare the input data for our algorithm. Thus the choice of the particular methods is not central to the topic of our study. In some cases, in particular for multiple sequence alignment and phylogenetic tree building, we selected the methods based on their efficiency. Due to the scale of the analysis and limited computational resources, we had to resign from using some algorithms which are considered more accurate, but are also considerably slower. Some of these alternative choices are mentioned in the text below. For a broader coverage of biological sequence analysis see Durbin *et al.* (1998) or Pevzner (2000) on which the following short review was partially based. Part of the material on sequence alignment and the neighbor joining algorithm also follows Tiuryn (2006).

2.1 Pairwise sequence alignment

Let σ be an alphabet not containing the symbol $'-'$ and let $\sigma' = \sigma \cup \{'-'\}$. Let S_1, S_2 be two sequences over σ, where $n = |S_1|$ and $m = |S_2|$ are lengths of these sequences, respectively. In our application the alphabet σ is the set of aminoacids which proteins are composed of.

2.1.1 Global alignment (Needleman-Wunsch)

A global alignment of sequences S_1, S_2 is given by two sequences S_1^A, S_2^A of length k ($k \leq n, m$) over the alphabet σ'. S_1^A is the sequence S_1 with the symbol $'-'$ inserted in $k - n$ positions. Similarly S_2^A is the sequence S_2 with the symbol $'-'$ inserted in $k - m$ positions.

Our goal is to find the highest-scoring alignment, where the score of the alignment is defined as:

$$\sum_{i=1\ldots k} s(S_1^A(i), S_2^A(i)),$$

where $S_1^A(i)$ ($S_2^A(i)$) is the i-th letter in sequence S_1^A (S_2^A) and $s : \sigma' \times \sigma' \longrightarrow \mathbb{R}$ is a scoring function corresponding to the similarity of a pair of letters. It is easy to see that we can compute the optimal alignment score by means of dynamic programming based on the recurrence (Needleman & Wunsch, 1970):

$$v_{i,j} = \max \begin{cases} v_{i-1,j-1} + s(S_1(i), S_2(j)) \\ v_{i-1,j} + s(S_1(i), '-') \\ v_{i,j-1} + s('-', S_2(j)) \end{cases},$$

where $v_{i,j}$ is the maximal alignment score for the prefixes $S_1[1..i]$ and $S_2[1..j]$. We assume that $v_{0,0} = 0$. To find the actual alignment, we can keep pointers backwards, and reconstruct the path by backtracking.

In practice the scoring function s is given by a substitution matrix which reflects the frequency at which a given letter of the alphabet substitutes another one. Many such matrices for various types of biological sequences (protein, DNA, RNA) have been proposed in the literature (see Durbin *et al.*, 1998).

2.1.2 Local alignment (Smith-Waterman)

It is often the case that only fragments of genes or proteins are similar to each other. Thus in biological applications we are often more interested in finding two most similar substrings of given sequences than the global alignment. Formally, in local alignment we want to find the maximum score

$$v^* = \max\{w_{i,j} | i \leq n, j \leq m\},$$

where $w_{i,j}$ is the maximum global alignment score between some suffix of the sequence $S_1[1..i]$ and some suffix of the sequence $S_2[1..j]$. It is easy to see that to compute $w_{i,j}$ we can consider global alignments of the sequences $S_1[1..i]$, $S_2[1..j]$ with the special option of a "fresh start" at any point in the alignment (meaning that we do not count the alignment up to that point in the sequences). The following recurrence expresses this idea (Smith & Waterman, 1981):

$$w_{i,j} = \max \begin{cases} 0 \\ v_{i-1,j-1} + s(S_1(i), S_2(j)) \\ v_{i-1,j} + s(S_1(i), '-') \\ v_{i,j-1} + s('-', S_2(j)) \end{cases} .$$

2.1.3 BLAST

In application where large databases are queried or when we need to exhaustively compute pairwise distances for a large number of proteins, fast alignment heuristics are especially useful. One particularly popular choice is the BLAST algorithm (Altschul *et al.*, 1990), designed to find high-scoring local alignments quickly, without guaranteeing the optimality of the result. The basic idea behind the algorithm is to start with short intervals of very high-scoring alignments of length l, referred to as *seeds*. Each of these seeds is expanded and alignments with scores above a threshold T are identified. The statistical significance of each of these alignments is assessed based on the Gumbel extreme value distribution (EVD). An E-value corresponding to the expected number of sequences from a given set which score higher than the given alignment is computed (see Altschul *et al.*, 1990; Durbin *et al.*, 1998).

2.2 Multiple sequence alignment

Multiple sequence alignment, i.e. alignment of three or more sequences, is a common strategy used to find conserved sequence regions, represent protein families and prepare data for reconstructing phylogenetic trees. We consider the following formulation of the multiple alignment problem.

For a pairwise sequence alignment S_1^A, S_2^B we define the alignment score by:

$$\delta(S_1^A, S_2^A) = \sum_{i=1\ldots n} \delta(S_1^A(i), S_2^A(i)),$$

where δ now denotes the distance between two letters of the alignment instead of their similarity. We define the score of a multiple alignment of l sequences by:

$$\delta(S_1^A, \ldots, S_l^A) = \sum_{i<j} \delta(S_i^A, S_j^A),$$

the so called *sum-of-pairs* distance.

The problem of finding a multiple sequence alignment $(S_1^A, \ldots, S_l^A)$ minimizing the sum-of-pairs score has been shown to be NP-hard (Wang & Jiang, 1994; Just, 2001). A practical approach to the problem is to align the sequences progressively. CLUSTALW is one of the most popular programs based on this concept.

2.2.1 CLUSTALW

The CLUSTALW algorithm (Higgins *et al.*, 1994) for multiple sequence alignment is based on the following scheme:

1. Align each pair of sequences S_i, S_j $(i < j)$ either approximately (faster) or exactly by using dynamic programming. The number of matches in the alignment is divided by the length of the alignment and subtracted from 1 to get the percent of mismatches. The percent of mismatches (number of mismatches per site) is used to approximate the evolutionary distance between the two sequences.

2. Based on the computed distances, construct a phylogenetic tree using the neighbor joining algorithm (see next section). The root of this tree is selected in a way such that the mean distances between the root and the leaves on each side of the tree are approximately equal.

3. The sequences are aligned progressively at each node (starting from the leaves and going up the tree), using sequence-to-sequence, sequence-to-profile and profile-to-profile alignments. The profiles represent alignments from previous stages of the algorithm. At each node the aligned instances are weighted according to their distance from this node in a way such that the most distant ones have the highest impact on the alignment score. For aligning profiles, the generalized score of aligning a column of symbols $x_1, \dots x_{r_1}$ with the column $y_1, \dots y_{r_2}$ takes the form:

$$\sum_{i=1..r_1} \sum_{j=1..r_2} s(x_i, y_j) w_{s_i} w_{s_j},$$

where x_i (y_j) comes from the sequence S_i (S_j) and w_{s_i} (w_{s_j}) is the weight of the sequence S_i (S_j).

CLUSTALW performs alignments heuristically and includes many additional steps that improve its performance – see Durbin *et al.* (1998) for an overview and Higgins *et al.* (1994) for details. Other popular methods for multiple sequence alignment include T-coffee (Notredame *et al.*, 2000) and Muscle (Edgar, 2004).

2.3 Sequence clustering

Sequence clustering is a primary tool for the identification of protein families. It many cases, it is desirable to cluster proteins from the entire proteomes of many species. These datasets typically contain hundreds of thousands of sequences. Thus again, the efficiency of the algorithm is crucial to its applicability. Another problem, which diminishes the performance of some more standard methods, is the presence of multi-domain and fragmented proteins. One of the recently developed algorithms that has gained significant recognition in the community is the Markov Cluster (MCL) algorithm by Enright *et al.* (2002). It has been successfully applied to cluster large databases of protein sequences (Enright *et al.*, 2002; Li *et al.*, 2003; Dujon & Sherman, 2004). Among other applications, MCL has also shown its effectiveness in detecting dense clusters in protein-protein interaction networks (Brohee & van Helden, 2006).

2.3.1 Markov Cluster

The MCL algorithm is designed to identify dense subgraphs (clusters) in large weighted graphs (Enright *et al.*, 2002). Intuitively, a graph cluster should have many edges between its members and relatively few to nodes outside. The weights of edges inside the cluster should generally be higher than those between clusters. Speaking in terms of random walks on graphs, we should rarely exit a natural cluster ones we are inside it.

MCL represents the similarity graph in the form of a column stochastic matrix, i.e. a non-negative matrix in which every column sums up to 1. The algorithm simulates random walks in the graph by alternating two matrix operations called *expansion* and *inflation*. Expansion coincides with matrix squaring (using normal matrix product). The inflation step involves taking the I-th power of every entry (Hadamard power) of the matrix and scaling the matrix to make it stochastic again. Expansion corresponds to computing random walks and is responsible for spreading out the stochastic flow in the graph. Longer paths should be more common within natural clusters, so the probabilities associated with transition to nodes within the same cluster should, in general, become higher. The inflation step further increases the higher probabilities and thus should have the effect of depressing the transitions between natural clusters. The parameter I controls the granularity of the clustering – larger I results in tighter clusters. Iterating the expansion and inflation steps results in a separation of the graph into separate connected components, identified as clusters. The global convergence of the algorithm has not been proven. In general convergence is noticeable after 3 to 10 iterations (Enright *et al.*, 2002).

The similarity relationships between proteins can be naturally represented in the form of a graph. Following Enright *et al.* (2002) we assign the weights in the graph based on the BLAST E-values. Specifically, since E-values are not necessarily symmetric, we take the average of the scores $-\log_{10}(E\text{-value})$ for the two proteins. This results in a symmetric matrix which is scaled to transform the weights to transition probabilities. MCL is applied to this matrix to identify protein families.

2.4 Phylogenetic trees

Phylogenetic trees are used to represent evolutionary history of species and biological sequences (genes and proteins). Trees which represent the evolution of species, referred to as *species trees*, are usually constructed from representative sequences (one from each species). To represent the evolution of sequences from a certain gene/protein family, a *gene family tree* (also called a *gene tree*) can be constructed. Gene trees may contain multiple sequences from one species – a result of possible gene duplication events. Phylogenetic tree reconstruction from sequence data is a fundamental problem in computational biology. Here we describe the neighbor joining algorithm which is efficient and provides a correct tree under certain conditions. Other popular methods have been developed based on optimization principles such as maximum likelihood and maximum parsimony (for a review see Nei, 1996). In section 2.4.2, we consider the task of reconciling a gene family tree with a species tree, consequently identifying protein speciation, duplication and loss events.

2.4.1 Neighbor joining

The neighbor joining (NJ) algorithm is a metric-based method for tree reconstruction. Suppose we are given a distance matrix $\{d_{ij}\}_{i,j=1..n}$, containing pairwise distances between n sequences – leaves of a tree T which is not provided. We assume that the distances are from a proper metric function, i.e. all of the following conditions are met for all i, j, k:

- $d_{ij} \geq 0$ and $d_{ij} = 0$ if and only if $i = j$,
- $d_{ij} = d_{ji}$,
- $d_{ij} \leq d_{ik} + d_{kj}$.

Let us further assume that the distances $\{d_{ij}\}_{i,j=1..n}$ are additive with respect to T, i.e. for every i and j, d_{ij} is equal to the sum of edge lengths on the path from i to j in T. Take two leaves i, j that are neighbors in T having (the same) parent p. Due to additivity, we have for each $m \neq i, j$:

$$d_{pm} = \frac{d_{im} + d_{jm} - d_{ij}}{2}, \tag{2.1}$$

where d_{vm} is the distance from a node v to m in T. Notice that we can now replace the two leaves by their parent p and consider a smaller problem. Following this way we can iteratively identify all distances in the tree. We only need to show how to identify the neighbors in the tree T from the distance matrix alone.

Notice that the choice of leaves with minimal distance to each other does not guarantee that they have the same parent – for an example see Durbin *et al.* (1998), p. 170. Instead, we define:

$$D_{ij} = d_{ij} - (r_i + r_j), \tag{2.2}$$

where r_i is defined as

$$r_i = \frac{1}{n-2} \sum_{k \in L} d_{ik}, \tag{2.3}$$

where L is the set of all leaves.

Theorem 2.1 *If a tree T has at least three leaves and $\{d_{ij}\}$ is additive with respect to T then each pair of leaves i, j which minimizes D_{ij} (2.2) is a pair of neighbors in T (Studier & Keppler, 1988).*

We can now write down the algorithm for reconstructing T from the matrix $\{d_{ij}\}$ (the NJ algorithm).

- Initialize L to be the set of leaves.
- Iterate:
  1. Select $i, j \in L$ with the minimal D_{ij} and remove them from L.
  2. Define a node p and set $d_{pm} = \frac{d_{im}+d_{jm}-d_{ij}}{2}$, for all $m \in L \setminus \{i, j\}$.
  3. The tree rooted at p is constructed from trees rooted at i and j by adding edges from p to i (j) with lengths $d_{ip} = \frac{d_{ij}+r_i-r_j}{2}$ ($d_{jp} = \frac{d_{ij}+r_j-r_i}{2}$).
- Stop when L has only two nodes u, v with the corresponding trees T_u and T_v and add the remaining edge with length d_{uv}. This way we receive an unrooted tree T.

Theorem 2.1 and equation (2.1) imply that the definition of d_{ip} gives the correct lengths. The theorem by Buneman (1971) provides a test to determine if the metric $\{d_{ij}\}$ is additive.

Theorem 2.2 *There exists a tree T for which the metric $\{d_{ij}\}$ is additive if and only if for every four leaves i, j, k and l, two of the distances $d_{ij} + d_{kl}$, $d_{ik} + d_{jl}$, $d_{il} + d_{jk}$ are equal and larger than the third (the four-point condition).*

In practice the NJ algorithm is often used when the distances are not additive, but without the guaranteed correctness.

2.4.2 Reconciliation with a species tree

We now consider the problem of reconciling a *gene family tree* with a given *species tree*. An intuitive illustration of the problem and motivations are given in Fig. 2.1. Below we formally define the reconciliation tree following Górecki & Tiuryn (2006a) and comment on the related problems and available algorithms.

Let S be a rooted species tree, i.e. a binary rooted tree whose leaves are uniquely labeled with species names from a set L. Let G be a rooted gene tree, i.e. a binary rooted tree whose leaves are uniquely labeled with gene names where each gene comes from one of the species in L (note that we can have multiple genes from the same species). In case of a species tree, the leaves are naturally associated with species. For gene trees we will associate each leaf with the species from which the gene comes from. For any tree T we denote by $T(v)$ the tree rooted at node v and by $L(T)$ the set of species associated with the leaves of T. We assume that the considered gene and species trees are non-empty.

For each node $g \in G$ let $M(g)$ be the node $s \in S$ for which

$$L(S(s)) = \bigcap \{L(S(w)) | w \in S \text{ and } L(G(g)) \subseteq L(S(w))\}.$$

In other words, each node g of the gene tree G is mapped by M to the *least common ancestor* in S of all the species associated with its leaves. In particular, each leaf of the gene tree is mapped to the leaf in S representing the same species from which the gene in the leaf comes from.

We now define the reconciled tree $R(G, S)$ (see Page & Charleston, 1997; Górecki & Tiuryn, 2006a). Let s be the root of S and g be the root of G. If G

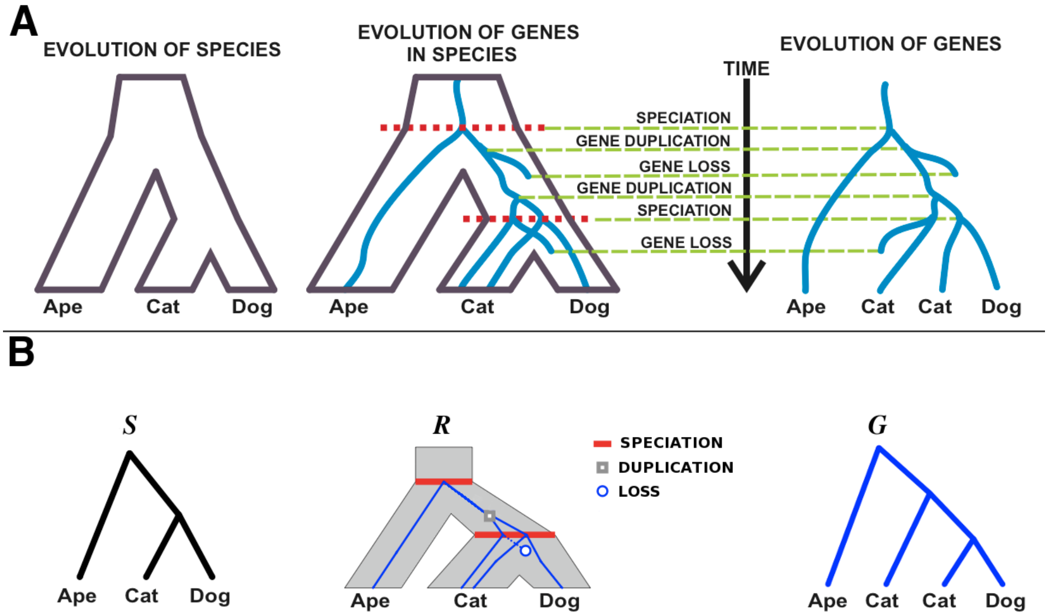


Figure 2.1: *Modeling evolutionary scenarios using embedded gene family trees. Part (A) illustrates a species tree (left) and a gene tree (right) representing the evolution of a family of genes, where each gene is labeled by the species from which the sequence was obtained. The evolution of the gene family is inherently tied with the evolution of the species – represented by the embedding of gene tree within the species tree (center). In (B) given a species tree (S) and a gene tree (G) reconstructed from sequence data, we aim to find evolutionary scenarios which explain the differences between them in terms of gene duplications and losses. One of such scenarios is represented by the reconciled gene tree (R) shown in the middle. It presents a biologically plausible case (preserving the original gene relationships) and is optimal in terms of gene duplications and losses. The internal nodes of the reconciled tree are related either to speciations, gene duplications or losses. Figure adapted from Górecki & Tiuryn (2006a).*

and S are both trees with only one leaf then $R(G,S) = G$. Else let p, q be the children of g. Then we have

$$R(G,S) = \begin{cases} (R(G(p),S), R(G(q),S)) & \text{if } M(g) = s = M(q) \\ (R(G(p),S), R(G(q),S)) & \text{if } M(g) = s = M(p) \\ (R(G(p),S(a)), R(G(q),S(b))) & \text{if } M(g) = s \text{ and } M(p) \in S(a) \\ & \text{and } M(q) \in S(b) \\ (R(G,S(a)), S(b)) & \text{if } M(g) \in S(a)\ b \neq a \end{cases}$$

where a and b are each children of s.

$R(G,S)$ represents an embedding of the tree G in the tree S (see Fig. 2.1). It is obvious that it can be computed in time $O(|G| * |S|)$. The tree $R(G,S)$ minimizes the number of duplication and speciation events over all scenarios that are feasible given the tree G and S (see Górecki & Tiuryn, 2006a). We will be interested in the case when the gene tree is unrooted (the NJ algorithm produces such trees). In this situation we can consider the number of duplication and speciation events as a criterion for choosing the correct root assignment. This kind of a parsimonious approach is often considered biologically relevant. Chen & Durand (2000) sketch a dynamic programming algorithm for identifying the rootings (possibly not unique) which minimize the sum of the number of duplication and speciation events. Górecki & Tiuryn (2006b) provide an algorithm (an prove its correctness) for the same problem, additionally taking into account positive weights for losses and duplications.

Chapter 3

Random graph models of network evolution

Properties of complex systems have always interested researchers, especially physicists and mathematicians. In the past years many efforts have been devoted to the study of large networks, in particular from the areas of computer science, engineering, economics and sociology. Recently, the development of efficient experimental techniques for identifying PPIs has opened a new domain for theoretical network studies. Large protein-protein interaction maps are analyzed to characterize their topological features and organisation. Through the construction and analysis of mathematical models one can hopefully gain understanding of how some the observed network properties might have emerged.

Networks are naturally modeled by graphs. Depending on the network domain, these graphs may be directed or have weighted edges. When modeling PPI networks, we will be mostly interested in undirected and unweighted graphs, possibly containing loops (edges which join nodes to themselves). We start this chapter with a review of some of the observed properties of protein interaction networks. Afterwards, we survey selected random graph models with respect to their applicability towards modeling the interactome.

3.1 Properties of PPI networks

Given two observed networks, a natural question to ask is if the networks are similar or if they contain isomorphic subnetworks. This question cannot be addressed effectively due to the hardness of the subgraph isomorphism problem. Instead, topological graph properties are commonly used to characterise and compare observed networks.

3.1.1 Degree distribution

One of the main, often characterised, network traits is the degree distribution. In many naturally occurring networks, in particular those of protein interactions, this distribution has been observed to closely resemble the power-law (see Yook *et al.*, 2004):

$$P(k) \propto k^{-\alpha},$$

where $P(k)$ is the fraction of nodes of degree k and α is a parameter (often taking value between 2 and 3). Networks having a power-law degree distribution are called *scale-free*.

3.1.2 Clustering coefficient

The clustering coefficient is often used to quantify the tendency for local cliques in the network. The clustering coefficient for a node v in the graph G is expressed as (Watts & Strogatz, 1998):

$$C_v(G) = \frac{\text{number of edges between neighbors of } v}{\binom{d_G(v)}{2}},$$

where $d_G(v)$ is the degree of the node v. The denominator denotes the number of pairs of edges incident to v. There are two alternative definitions for the clustering coefficient of the entire graph G. The one most often used corresponds to the mean clustering coefficient over all nodes:

$$C^{(1)}(G) = \sum_{v=1}^{n} C_v(G)/n.$$

In the alternative definition Bollobás (2003) takes a weighted mean considering the degree of each node:

$$C^{(2)}(G) = \left(\sum_{v=1}^{n} \binom{d_G(v)}{2} C_v(G) \right) \Big/ \sum_{v=1}^{n} \binom{d_G(v)}{2}.$$

This definition is equivalent to:

$$C^{(2)}(G) = \frac{\text{number of pairs of adjacent edges } ab,\ ac, \text{ for which } bc \text{ is an edge}}{\text{number of pairs of adjacent edges } ab,\ ac}.$$

It has been postulated that PPI networks have a constant positive clustering coefficient which is independent of the number of nodes. Often it is simply described as larger than in random graphs (in the sense of Gilbert's model, see Section 3.2.1).

3.1.3 Diameter

The distance between two nodes in the network is defined as the shortest path between them. The diameter of a network is the longest distance between any two nodes. Observed PPI networks are usually composed of a number of connected components of which one is dominating in size and is referred to as the *giant component*. The diameter of the observed giant components is usually of the order of $\log n_c$, where n_c is the number of nodes.

3.2 Selected models of random graphs

3.2.1 Gilbert's model

The model of Gilbert (1959) (see also Bollobás, 2001; Bollobás, 2003) is the space $G_{n,p}$ of random graphs with n vertices, in which each edge $v_i v_j$ $(i \neq j)$ occurs with probability p. In other words, if G is a graph from this space with m edges then the probability of G is

$$P(G) = p^m (1-p)^{N-m},$$

where $N = \binom{n}{2}$. Gilbert's model is the most widely studied and characterised random graph model. In many cases it is interchangeable with a slightly different model studied by Erdös & Rényi (1959). Below we list some of the known results for $n \longrightarrow \infty$ and $p = c/n$, where $c > 0$ is a constant.

Degree distribution The degree distribution in this model is approximated by the Poisson distribution (see Bollobás, 2003):

$$Pr\left((1-\epsilon)\frac{c^k e^{-c}}{k!} \leq P(k) \leq (1+\epsilon)\frac{c^k e^{-c}}{k!}\right) \longrightarrow 1,$$

for $\epsilon > 0$.

Clustering coefficient The expected value of the clustering coefficient is $O(n^{-1})$ for both definitions in Section 3.1.2.

Expected number of subgraphs isomorphic to a given graph Let H be a graph with k vertices and $l \geq 2$ edges. By $Aut(H)$ we denote the automorphism group of H. Denote by X the number of subgraphs isomorphic to H in the random graph G from the space $G_{n,p}$. Then the expected value of X is given by:

$$\mathbb{E}(X) = \binom{n}{k}\frac{k!}{|Aut(H)|}p^l \sim \frac{n^k}{|Aut(H)|}p^l.$$

It is also possible to show that the distribution of subgraphs isomorphic to H converges to Poisson distribution, if H is a graph which is strictly balanced (see Bollobás, 2001).

Unfortunately, due to its degree distribution and the declining clustering coefficient, this simple model is not appropriate for PPI networks.

3.2.2 Preferential attachment

In recent years many models have been proposed to explain the scale-free nature of naturally occurring networks. Often these models were stated in natural language, not very formally, and the postulated properties were supported only by simulations. One of the most popular "scale-free" models is a model by Barabasi & Albert (1999) which is based on the principle of *preferential attachment*. Bollobás *et al.* formalized the definition of the model and provided rigorous analysis of some of its properties (Bollobás *et al.*, 2001; Bollobás, 2003; Bollobás & Riordan, 2004). They use the *linearized chord diagrams* in the model analysis, hence the model name LCD.

LCD model Let $v_1, v_2, \ldots$ be a sequence of vertices. We define by induction a random graph process $(G_m^{(t)})_{t \geq 0}$, such that $G_m^{(t)}$ is a graph with vertices $\{v_i : 1 \leq i \leq t\}$ and m edges exiting each vertex. Note that the edges are described as directed, although this is in fact not essential to the model and the proven properties. Each edge contributes to the degree of both adjacent nodes (adds one to both degrees). We first analyze the case when $m = 1$:

- We start with $G_1^{(0)}$ (an empty graph) or $G_1^{(1)}$ (a graph with one vertex and one edge).
- The graph $G_1^{(t)}$ is constructed from the graph $G_1^{(t-1)}$ by adding vertex v_t and one edge between v_t and v_i where i is selected such that

$$Pr(i = s) = \begin{cases} d_{G_1^{(t-1)}}(v_s)/(2t-1) & 1 \leq s \leq t-1, \\ 1/(2t-1) & s = t. \end{cases}$$

Thus vertex v_s is selected with probability proportional to its degree, according to the preferential attachment principle. We assume that a new edge exiting v_t adds 1 to its degree. For $m > 1$, we add m edges exiting v_t, one by one, each time updating the degrees of the vertices.

Degree distribution Bollobás *et al.* (2001) show that in the limit the degree distribution follows a power-law with parameter $\alpha = 3$.

The expected number of triangles Bollobás (2003) shows that the expected number of triangles (3 - element cliques) in random graph from this model is asymptotically equal to

$$\mathbb{E}(\#\triangle) \sim \frac{m(m-1)(m+1)}{48}(\log n)^3,$$

when $n \longrightarrow \infty$.

Clustering coefficient Based on the above result, Bollobás (2003) shows that the expected value of the clustering coefficient (by the second definition in Section 3.1.2) is asymptotically equal to

$$\mathbb{E}(C^{(2)}(G_m^{(n)})) \sim \frac{m-1}{8}\frac{(\log n)^2}{n},$$

for $n \longrightarrow \infty$.

Diameter Bollobás & Riordan (2004) show that for $m \geq 2$ and a constant $\epsilon > 0$, with probability tending to 1, $G_m^{(n)}$ is connected and has the diameter $(\mathrm{diam}(G_m^{(n)}))$ satisfying

$$(1-\epsilon)\log n/\log\log n \leq \mathrm{diam}(G_m^{(n)}) \leq (1+\epsilon)\log n/\log\log n.$$

The model's most significant limitations with respect to modeling PPI networks lie in the incapability to fit the degree distribution to data (the power-law parameter $\alpha = 3$) and the diminishing clustering coefficient. Furthermore, the preferential attachment rule cannot be easily interpreted in terms biological processes driving PPI network evolution.

3.2.3 Duplication and divergence

We now draw attention to some biological considerations. These are important if we want to interpret the model in terms of the underlying mechanisms which are responsible for producing network topologies observed in nature. The most important high-level events which impact the PPI network are: protein duplication, protein deletion, interaction emergence and interaction deletion.

Protein duplications are a crucial factor responsible for the growth of the network. Furthermore, duplications provide an opportunity for emergence of new functionality, since the system does not usually need two proteins with exactly the same function (Ohno, 1970). It is often assumed that protein deletion events are rare and so their impact is less considerable. Given the above, we now describe a model proposed by Sole *et al.* (2002) (see also Pastor-Satorras *et al.*, 2003).

Model definition We define a random graph process $(G^{(t)}_{n_0,p,r})_{t\geq 0}$, such that $G^{(t)}_{n_0,p,r}$ is an undirected graph with $n_0 + t$ vertices.

- Start with an undirected connected graph with n_0 vertices.
- Iterate the following steps:
  - **Duplication**: At time t a vertex w is drawn uniformly at random from the set of vertices in the graph $G^{(t-1)}_{n_0,p,r}$ and copied forming the vertex v_t (v_t is initially connected to all neighbors of w).
  - **Divergence**: Next, we modify the edges incident to v_t:
    1. each edge incident to v_t is considered independently and removed with probability $q = \ (1-p)$,
    2. each node u which was not connected to v_t (before the last step) is considered independently and an edge v_tu is added with probability r.

Degree distribution Bebek *et al.* (2006) postulate that for $r > 0$ and $k \geq 1$ the fraction of vertices of degree k obeys

$$P(k) = (1 + O(1/k))ck^{-b},$$

where b is the solution of the equation $1 = pb - p + p^{b-1}$.

Experimental results of Bebek *et al.* (2005) suggest that the clustering coefficient is close to the one in the observed PPI networks. The parameters of the model provide the possibility of fitting the degree distribution to the observed data. The duplication and divergence model has a definite advantage – it is biologically motivated. Similar models have been proposed and analysed by other authors. Ispolatov *et al.* (2005b) analyse the case in which no new links are introduced (r = 0) and nodes without any interactions are removed. They find that the model generates graphs which are very similar to the ones observed in nature. They study the average vertex degree depending on the conservation of links in the newly duplicated protein. They conclude that the average degree increases very slowly or tends to a constant when the link conservation is low. In contrast, when the conservation is high, the network growth is not self-averaging and results in a diversity between grown networks. Ispolatov *et al.* (2005a) study a version of the model which includes heterodimerization links between duplicates. They postulate that the model correctly describes the clique statistic observed in natural PPI networks. They also consider symmetric models in which the divergence can occur in both copies of the duplicated protein – the previously present and the newly added one. This scenario is supported by the findings of Kondrashov *et al.* (2002) who show that duplicate proteins typically evolve at similar rates and both copies are subject to purifying selection. Symmetric version of the duplication and divergence model has also been investigated by Vazquez *et al.* (2003).

3.2.4 Final remarks

Other models for PPI networks have also been considered. Przulj *et al.* (2004) suggest that a geometric random graph model (see Penrose, 2003) better fits the observed networks than the preferential attachment scale-free models. The authors use a new measure of local network structure which is based on counting the occurrences of small subgraphs (graphlets). More recently, the graphlet distribution, among other characteristics, was used to show that the networks grown by the duplication and divergence scale-free model (in contrast to the preferential-attachment model) correspond well to the observed PPI data when the growth process is initiated from dense seed graphs (Hormozdiari *et al.*, 2007).

An interesting and decisive example of how careful one must be when claiming that a given model fits the observed data and produces graphs with desired properties was given by Bollobás (2003). Bollobás demonstrated that the imprecise definition of Barabasi & Albert (1999) leads to graphs of very diverse nature, depending on the small graph from which the growth process is initiated. Similar observations were recently made by Hormozdiari *et al.* (2007). It is also worth to remember that while most formally analyzed properties of random graph models are proven asymptotically for the number of nodes going to infinity, the naturally occurring networks are of finite size (captured in a given stage of evolution). Thus their characteristics might not be directly comparable. Finally, it is also possible that many rather different models may provide graphs matching some or all of the topological characteristics of interest (once the model parameters are fitted to the data).

In our analysis, we assume that PPI networks evolve under a model similar in spirit to the duplication and divergence models of Sole *et al.* (2002), as it is biologically relevant and provides plausible explanations for the traits observed in natural networks. Our model, presented in Chapter 5, assumes that the duplication events (and also speciations) are determined by pre-computed phylogenies. We assume symmetric divergence of duplicates and consider possible interactions between them. Based on this model we develop an inference framework that employs available experimental data to reconstruct ancestral states of the interactome and predict missing interactions in present-day species.

Chapter 4

Bayesian network models and inference

In this chapter we review Bayesian network (BN) models and discuss one of the primary problems addressed using this framework, namely Bayesian inference.

4.1 Bayesian networks

We start by defining the Markov condition, following Neapolitan (2003). Let V be a set of random variables. Let P be the joint probability distribution of these random variables. Finally, let $G = (V, E)$ be a directed acyclic graph (DAG) in which the nodes are identified by the random variables in V. For a pair of nodes $X, Y \in V$ $(X \neq Y)$, X is called a *parent* of Y if there is an edge in E from X to Y. Y is called a *descendant* of X if there exists a path from X to Y in G. Otherwise Y is a *nondescendant* of X. The pair (G,P) satisfies the *Markov condition* if for each variable $X \in V$, $\{X\}$ is conditionally independent of the set of all its nondescendants given the set of all its parents. A pair (G,P) which satisfies the Markov condition is called a *Bayesian network*. The joint distribution P is represented in a Bayesian network by the set of conditional distributions of each node given the values of its parents. The following theorem provides means for easy computation of the joint probability from the conditional probabilities between neighbors in G (see Neapolitan, 2003, p. 39):

Theorem 4.1 *If (G, P) satisfies the Markov condition (i.e. is a Bayesian network), then P is equal to the product of its conditional distributions of all nodes given values of their parents, whenever these conditional distributions exist.*

The next theorem establishes the basis for easy construction of Bayesian networks (see Neapolitan, 2003, p. 42):

Theorem 4.2 *Let G be a DAG in which each node is a random variable and let a discrete conditional probability distribution of each node given the values of its parents be specified. Then the product of these distributions yields a joint probability distribution (let us call it P), and (G, P) satisfies the Markov condition.*

As we will work only with discrete distributions, the above theorem is sufficient for us. We note however, that many continuous conditional distribution specified for G also yield a Bayesian network (see Neapolitan, 2003).

An important concept in Bayesian networks is that of *d-separation* which implies conditional independence of separated nodes. We introduce useful definitions and state the theorem, again following Neapolitan (2003).

A *chain* is a sequence of edges from the set E which defines a path in the undirected graph underlying the DAG G. We consider three types of meetings of edges on a chain ρ:

- A *head-to-tail meeting* at Z is a meeting of type $X \rightarrow Z \rightarrow Y$, where $X \rightarrow Z, Z \rightarrow Y \in E$ are edges on the chain ρ.

- A *tail-to-tail meeting* at Z is a meeting of type $X \leftarrow Z \rightarrow Y$, where $X \leftarrow Z, Z \rightarrow Y \in E$ are edges on the chain ρ.

- A *head-to-head meeting* at Z is a meeting of type $X \rightarrow Z \leftarrow Y$, where $X \rightarrow Z, Z \leftarrow Y \in E$ are edges on the chain ρ.

Let $A \subseteq V$ be a subset of nodes of a DAG G and $X, Y \in V \setminus A$ be distinct nodes. Let ρ be a chain between X and Y. We say ρ is *blocked* by A if one of the following holds:

- There is a node $Z \in A$ on the chain ρ, at which the edges on ρ meet head-to-tail.

- There is a node $Z \in A$ on the chain ρ, at which the edges on ρ meet tail-to-tail.

- There is a node Z on the chain ρ, at which the edges on ρ meet head-to-head and Z is not in A and none of Z's descendants are in A.

Note that the third condition may hold also if $A = \emptyset$. We say that X and Y are *d-separated* by $A \subseteq V$ if every chain between X and Y is blocked by A. Given three mutually disjoint subsets of nodes $A, B, C \subseteq V$, we say that A and B are *d-separated* by C if for every $X \in A$ and every $Y \in B$, X and Y are d-separated by C. Note again that C may also be an empty set. The following theorem establishes the implication of conditional independence from d-separation in Bayesian networks (see Neapolitan, 2003, p. 77):

Theorem 4.3 *Let P be a probability distribution of the variables in V and $G = (V, E)$ be a DAG. Then (G, P) satisfies the Markov condition if and only if, for every three mutually disjoint subsets $A, B, C \subseteq V$, whenever A and B are d-separated by C, A and B are conditionally independent in P given C.*

If $C = \emptyset$ then A and B are independent in P.

4.2 Inference

We now consider the problem of *probabilistic inference*, i.e. computing the conditional probability distribution of a random variable given the observed values of one or more other random variables. We assume that the joint probability distribution is provided by a Bayesian network, which will enable us to take advantage of the entailed conditional independencies. In general, we will be interested in computing all posterior marginals of non-instantiated variables given the instantiated variables (evidence). We start with a few examples of inferences in a discrete Bayesian network shown in Figure 4.1. We assume that each variable can take value either 1 or 0 (i.e. is a binary random variable). The joint distribution P is given by the local conditional probability tables and by the *prior* probability at node X.

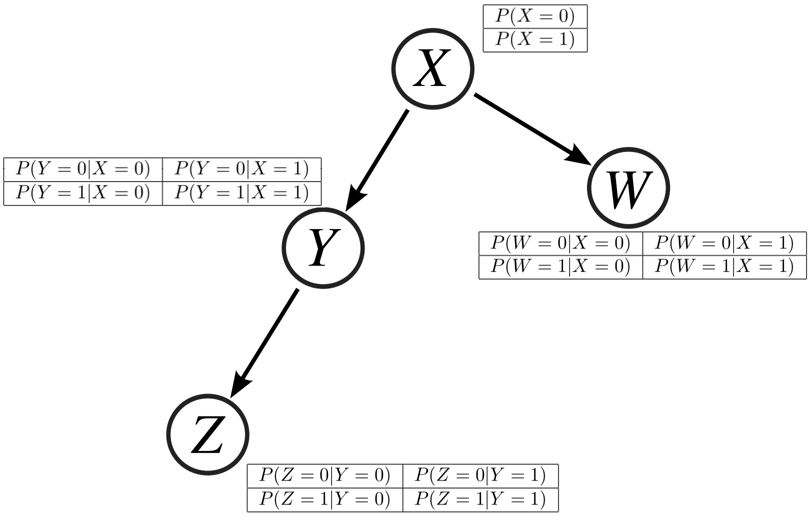

$P(X=0)$
$P(X=1)$
X
$P(Y=0|X=0)$ $P(Y=0|X=1)$
$P(Y=1|X=0)$ $P(Y=1|X=1)$
W
Y
$P(W=0|X=0)$ $P(W=0|X=1)$
$P(W=1|X=0)$ $P(W=1|X=1)$
Z
$P(Z=0|Y=0)$ $P(Z=0|Y=1)$
$P(Z=1|Y=0)$ $P(Z=1|Y=1)$

Figure 4.1: *An example of a Bayesian network with four binary random variables. The local conditional probability tables specify the probability distribution of a node given its parent. For the node X the prior probability distribution is given.*

Let's start by computing the *prior* probabilities (i.e. without assuming any values of other variables):

$$\begin{aligned}
P(W=1) &= P(W=1|X=1)P(X=1) + P(W=1|X=0)P(X=0)\\
P(Y=1) &= P(Y=1|X=1)P(X=1) + P(Y=1|X=0)P(X=0)\\
P(Z=1) &= P(Z=1|Y=1)P(Y=1) + P(Z=1|Y=0)P(Y=0),
\end{aligned}$$

where in the last equation we can reuse the probability distribution $P(Y)$ computed in the second equation. Let us know consider simple inferences using the Bayes' theorem:

$$\begin{aligned}
P(X=1|W=1) &= \frac{P(W=1|X=1)P(X=1)}{P(W=1)}\\
P(X=1|Y=1) &= \frac{P(Y=1|X=1)P(X=1)}{P(Y=1)}\\
P(Y=1|Z=1) &= \frac{P(Z=1|Y=1)P(Y=1)}{P(Z=1)},
\end{aligned}$$

where we can again use the previously computed values of the *prior* probabilities. Let us do one final inference in this network and compute the probability $P(W=1|Z=1)$:

$$\begin{aligned}
P(W=1|Z=1) &= \sum_{x\in\{0,1\}} P(W=1, X=x|Z=1) && (4.1)\\
&= \sum_{x,\in\{0,1\}} P(W=1|X=x, Z=1)P(X=x|Z=1) && (4.2)\\
&= \sum_{x,\in\{0,1\}} P(W=1|X=x) \sum_{y\in\{0,1\}} P(X=x, Y=y|Z=1) && (4.3)\\
&= \sum_{x\in\{0,1\}} P(W=1|X=x) \sum_{y\in\{0,1\}} P(X=x|Y=y, Z=1)\\
&\cdot P(Y=y|Z=1) && (4.4)\\
&= \sum_{x\in\{0,1\}} P(W=1|X=x) \sum_{y\in\{0,1\}} P(X=x|Y=y)\\
&\cdot P(Y=y|Z=1). && (4.5)
\end{aligned}$$

In (4.1) and (4.2) we introduced variable X and summed it out to break the dependency between W and Z. In (4.3) we took advantage of the conditional independence entailed by the Markov condition and we introduced and summed over the variable Y. In (4.4) and (4.5), due to d-separation, we were able to express $P(X = x, Y = y|Z = 1)$ using two local conditional probabilities (between neighbors) which can be computed using the Bayes' theorem and the previously derived prior probabilities. Similarly as before when we computed the prior probabilities incrementally and stored them for each node, here we can compute the conditional distribution $P(X|Z)$ using the conditional distribution $P(Y|Z)$ of the child. These ideas are used in the forward-backward algorithm (operating in the context of Hidden Markov Models) and by its generalization, Pearl's message passing algorithm, which we describe next.

4.3 Pearl's message passing algorithm

For general Bayesian networks the inference problem has been shown to be NP-hard (Cooper, 1990). We now derive Pearl's message passing (MP) algorithm (Pearl, 1988) for inference in singly connected discrete Bayesian networks. In singly connected graphs (also called polytrees) there is at most one undirected path between any two nodes. A Bayesian network is singly connected if its underlying graph is singly connected. Our notation and reasoning in this section follows Murphy (1999) and in parts also Neapolitan (2003).

Let (G, P) be a singly connected Bayesian network. Without the loss of generality, we assume that the underlying undirected graph is connected. Let $A \subseteq V$ be a set of evidence nodes i.e. variables for which we observe values encoded by a vector $\mathbf{a}$. We are interested in computing the *posterior* probability distribution (given the data) $P(X = x|A = \mathbf{a})$ for every node X. Let $N_X \subseteq A$ contain all evidence nodes that are above X, i.e. in the subgraph connected to X *via* one of its incoming edges (excluding X). Let $D_X = A - N_X$ contain all other evidence nodes, i.e. those that are below X (in the subgraph connected to X *via* one of its outgoing edges) and also X if $X \in A$. We write $\mathbf{n}_X$ and $\mathbf{d}_X$ for the particular instances of the observed random variables from N_X and D_X respectively. We also use a shorter notation writing $P(z)$ for $P(Z = z)$ where z

is one of the possible values taken by the random variable Z. Using the Bayes' theorem and the d-separation rule we have:

$$\begin{aligned} P(x|\mathbf{a}) &= P(x|\mathbf{d}_X, \mathbf{n}_X) = \\ &= \frac{P(\mathbf{d}_X, \mathbf{n}_X|x)P(x)}{P(\mathbf{d}_X, \mathbf{n}_X)} = \\ &= \frac{P(\mathbf{d}_X|x)P(\mathbf{n}_X|x)P(x)}{P(\mathbf{d}_X, \mathbf{n}_X)} = \\ &= \frac{P(\mathbf{d}_X|x)P(x|\mathbf{n}_X)P(\mathbf{n}_X)P(x)}{P(\mathbf{d}_X, \mathbf{n}_X)P(x)} = \\ &= \alpha P(\mathbf{d}_X|x)P(x|\mathbf{n}_X), \end{aligned} \tag{4.6}$$

where α is a constant independent of the value x.

4.3.1 λ and π values

We denote the two probabilities in (4.6) by:

$$\lambda_X(x) \overset{\text{def}}{=} P(\mathbf{d}_X|x) \tag{4.7}$$

$$\pi_X(x) \overset{\text{def}}{=} P(x|\mathbf{n}_X). \tag{4.8}$$

Our goal is to compute (4.7) and (4.8) efficiently for each X. Notice that to compute the unnormalized distribution of the form (4.6) it is enough to know $\beta\lambda_X(x)$ and $\gamma\pi_X(x)$, where β and γ are constants independent of x. These constants cancel out when we normalize the distribution. Let us consider the situation in Fig. 4.2. The figure shows a random variable X together with its parents $(U_1 \dots U_k)$ and children $(Y_1 \dots Y_l)$ in the graph. We write $\mathbf{n}_{U_i \to X}$ for the evidence in nodes above the edge $U_i \to X$ (including U_i if $U_i \in A$), and $\mathbf{d}_{X \to Y i}$ for the evidence below the edge from X to Y_i (including Y_i if $Y_i \in A$). We denote the possible evidence in X by e_X. We define local λ messages that will be sent from a node X to a parent node U_i as follows:

$$\lambda_{X \to U_i}(u_i) \overset{\text{def}}{=} P(\mathbf{d}_{U_i \to X}|u_i).$$

The evidence nodes in each subtree rooted in Y_i and the possible evidence in X are conditionally independent given X. Thus we can write:

$$\lambda_X(x) = \prod_{i=1\dots l} P(\mathbf{d}_{X \to Y i}|x) = \prod_{i=1\dots l} \lambda_{Y_i \to X}(x),$$

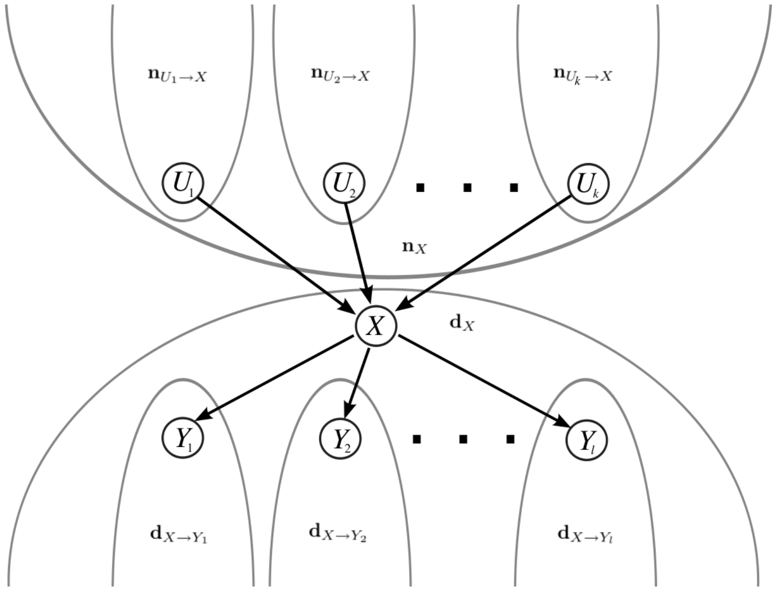


Figure 4.2: *Part of a Bayesian network centered around node X. The evidence below and in X is denoted by $\mathbf{d}_X$. The evidence above X is denoted by $\mathbf{n}_X$. The evidence above each edge going from a parent U_i to X is denoted by $\mathbf{n}_{U_i \to X}$ for $i = 1 \dots k$. The evidence below each edge going from X to a child node Y_i is given by $\mathbf{d}_{X \to Y_i}$ for $i = 1 \dots l$.*

if $X \notin A$, and

$$\lambda_X(x) = P(e_X|x) \prod_{i=1\dots l} P(\mathbf{d}_{X \to Yi}|x) = P(e_X|x) \prod_{i=1\dots l} \lambda_{Y_i \to X}(x),$$

if $X \in A$. If X is a leaf (i.e. has no children) and is not an evidence node then we assume that $\lambda_X(x) = 1$ for each x.

To compute $\pi_X(x)$ we consider the nodes upstream of X. Let $\mathbf{u} = u_1 \dots u_k$ denote a particular instance of the parent nodes $U_1 \dots U_k$. Summing over all possible values $\mathbf{u}$ and considering the conditional independencies we have:

$$P(x|\mathbf{n}_X) = \sum_{\mathbf{u}} P(x, \mathbf{u}|\mathbf{n}_X) = \sum_{\mathbf{u}} P(x|\mathbf{u})P(\mathbf{u}|\mathbf{n}_X) = \sum_{\mathbf{u}} \left[P(x|\mathbf{u}) \prod_i P(u_i|\mathbf{n}_{U_i \to X}) \right].$$

We define local π messages sent from a node X to each of its children as follows:

$$\pi_{X \to Y_i}(x) \overset{\text{def}}{=} P(x|\mathbf{n}_{X \to Yi}).$$

Using the defined messages and we can write:

$$\pi_X(x) = \sum_{\mathbf{u}} \left[P(x|\mathbf{u}) \prod_i \pi_{U_i \to X}(u_i) \right].$$

If X is a root (i.e. has no parents) we assume that $\pi_X(x) = P(x)$, the prior probability of x.

4.3.2 λ and π messages

Now computing $\lambda_X(x)$ and $\pi_X(x)$ boils down to computing the λ and π messages. We start with the λ message passed from a node X to one of its parents U_i. In this case we have to consider all the evidence except the evidence $\mathbf{n}_{U_i \to X}$ (see Fig. 4.3 for an example).

$$\lambda_{X \to U_i}(u_i) = P(\mathbf{d}_X, \mathbf{n}_{U_1 \to X}, \dots, \mathbf{n}_{U_{i-1} \to X}, \mathbf{n}_{U_{i+1} \to X}, \mathbf{n}_{U_k \to X}|u_i).$$

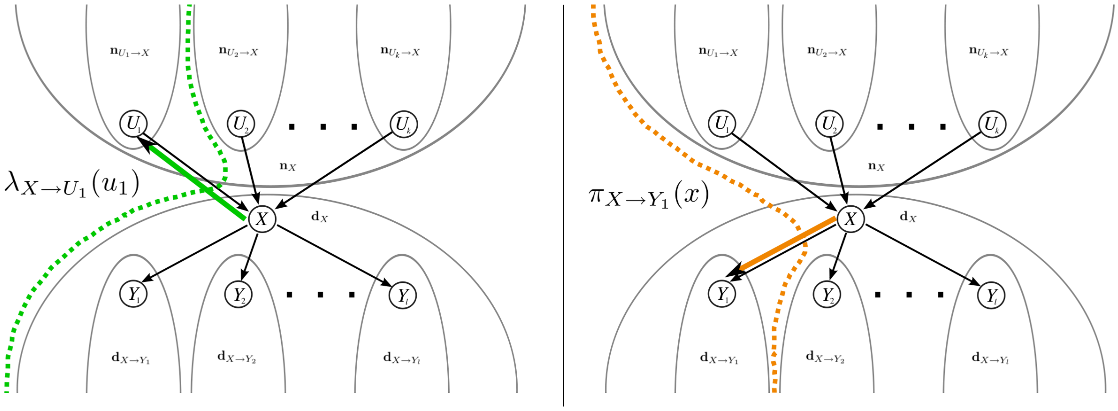


Figure 4.3: *An example of a λ message (left) and a π message message (right). The λ message from X to U_1 considers all evidence except the evidence above the edge from U_1 to X. The π message from X to Y_1 considers all evidence except the evidence below the edge from X to Y_1.*

We sum over all possible instances x and $\mathbf{u} - u_i \overset{\text{def}}{=} (u_1, \ldots, u_{i-1}, u_{i+1}, u_k)$ to break the dependencies:

$$\begin{aligned}
\lambda_{X\to U_i}(u_i) = &\sum_x \sum_{\mathbf{u}-u_i} P(\mathbf{d}_X, \mathbf{n}_{U_1\to X}, \ldots, \mathbf{n}_{U_{i-1}\to X}, \mathbf{n}_{U_{i+1}\to X}, \mathbf{n}_{U_k\to X} | u_1, \ldots, u_k, x) \\
&\cdot P(u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_k, x | u_i) \\
&= \sum_x \sum_{\mathbf{u}-u_i} P(\mathbf{d}_X|x) P(\mathbf{n}_{U_1\to X}|u_1) \cdots P(\mathbf{n}_{U_{i-1}\to X}|u_{i-1}) \cdot \\
&\cdot P(\mathbf{n}_{U_{i+1}\to X}|u_{i+1}) \cdots P(\mathbf{n}_{U_k\to X}|u_k) P(u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_k, x | u_i).
\end{aligned}$$

Applying the Bayes' theorem for every $j \neq i$ we obtain:

$$P(\mathbf{n}_{U_j\to X}|u_j) = \frac{P(u_j|\mathbf{n}_{U_j\to X}) P(\mathbf{n}_{U_j\to X})}{P(u_j)} = \beta_j \frac{P(u_j|\mathbf{n}_{U_j\to X})}{P(u_j)},$$

where β_j is a constant. We can also write

$$\begin{aligned}
&P(u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_k, x | u_i) = \\
&= P(x|u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_k, u_i) P(u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_k | u_i) \\
&= P(x|u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_k, u_i) P(u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_k) \\
&= P(x|u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_k, u_i) P(u_1) \cdots P(u_{i-1}) P(u_{i+1}) \cdots P(u_k)
\end{aligned}$$

since all U_j $(j = 1 \dots k)$ are marginally independent (head-to-head meeting at X of the respective chains implies d-separation). Putting it all together, we have:

$$\lambda_{X\to U_i}(u_i) = \beta \sum_x P(\mathbf{d}_X|x) \left[\sum_{\mathbf{u}-u_i} P(x|u_1,\dots,u_{i-1},u_{i+1},\dots,u_k,u_i) \prod_{j\neq i} P(u_j|\mathbf{n}_{U_j\to X}) \right],$$

where $\beta = \beta_1 \cdots \beta_{i-1}\beta_{i+1} \cdots \beta_k$ is a constant. We can express this in terms of the λ values and π messages as follows:

$$\beta \sum_x \lambda_X(x) \left[\sum_{\mathbf{u}-u_i} P(x|u_1,\dots,u_{i-1},u_{i+1},\dots,u_k,u_i) \prod_{j\neq i} \pi_{U_j\to X}(u_j) \right].$$

A special case, which we will be of interest to us, occurs when the underlying DAG is a rooted tree. Then node X has at most one parent U and we have

$$\lambda_{X\to U}(u) = \sum_x \lambda_X(x) P(x|u).$$

To compute the π message sent from node X to its child Y_i we have to consider all evidence except the evidence below the edge from X to Y_i (see Fig. 4.3). We can express this in terms of the value π_X and the λ messages coming from all children of X except Y_i:

$$\begin{aligned}
\pi_{X\to Y_i}(x) &= P(x|\mathbf{n}_{X\to Y_i}) = \frac{P(\mathbf{n}_{X\to Y_i}|x)P(x)}{P(\mathbf{n}_{X\to Y_i})} \\
&= \frac{P(\mathbf{n}_X|x)P(e_X|x)\prod_{j\neq i} P(\mathbf{d}_{X\to Y_j}|x)P(x)}{P(\mathbf{n}_{X\to Y_i})} \\
&= \frac{P(x|\mathbf{n}_X)P(\mathbf{n}_X)P(e_X|x)\prod_{j\neq i} P(\mathbf{d}_{X\to Y_j}|x)P(x)}{P(x)P(\mathbf{n}_{X\to Y_i})} \\
&= \gamma P(x|\mathbf{n}_X)P(e_X|x)\prod_{j\neq i} P(\mathbf{d}_{X\to Y_j}|x) = \gamma \pi_X(x)P(e_X|x)\prod_{j\neq i} \lambda_{Y_j\to X}(x),
\end{aligned}$$

if $X \in A$, and otherwise

$$\pi_{X\to Y_i}(x) = \gamma \pi_X(x) \prod_{j\neq i} \lambda_{Y_j\to X}(x),$$

where γ is a constant.

4.3.3 Message propagation order

The simplest case occurs when the underlying DAG is a rooted tree. Each node can send its λ message to its single parent once it gets the incoming λ messages from all its children. It is easy to see that we can start by passing λ messages from the leaves and go up the tree (visiting children before parents) until we reach the root. Once we reach the root, we can begin sending π messages starting from the root and moving in preorder (visiting the parents before we reach the children).

In a more general case, when the DAG is a polytree, we can select any node as the root of the tree in the undirected sense. The selection of the root determines a single parent for each node in the undirected graph (the only neighbor that is on the path from this node to the root). We can again start by sending messages from the leaves of such a tree towards the selected root. This time however, depending on the direction of the original edges, we might have to alternate the messages. We will send a λ message when we move against the direction of the original edge, and a π message when we move with the direction of the original edge. To send either a λ or a π message from a node X to a node U we need the information from all other neighbors (see Fig. 4.3). We traverse the undirected tree in postorder so each node will get the incoming messages from all its children (either λ or π messages) before it has to send an outgoing message either (λ or π) to its single parent. Again, once we get to the root, we switch to preorder tree traversal and send the opposite messages starting from the root and going towards the leaves.

4.3.4 Complexity of the MP algorithm

The running time of the described algorithm is linear in the number of nodes of the polytree, polynomial in the maximum number of values a node can take, and exponential in the maximum number of parents of a node. Let n be the number of nodes in the polytree, k – the maximum number of values a node can take, p – the maximum number of parents of each node, and c – the maximum number of children of each node. For each node, we have at most $k^{p+1}p^2$ multiplications needed to compute all λ messages, $kc(c+1)$ multiplications to compute all π messages, $k(c+1)$ multiplications to compute the λ value, $k^{p+1}(p+1)$ to compute the π value, and k multiplications to compute the posterior probability at the node. The total number of multiplications is thus $O(nk^{p+1}(p^2 + p + 1) + nkc^2 + n2kc + n2k)$.

Chapter 5

Network alignment: identifying conserved subnetworks

In this chapter we develop our network comparison framework. It is based on the reconstruction of a conserved ancestral protein-protein interaction (CAPPI) network, which lends the name to our method. First, we reconstruct hypothetical sequences of evolutionary events (duplications, speciations and deletions), by which the proteins of the input PPI networks evolved from their counterparts in the common ancestral network. In the next step, we determine the posterior probabilities of interaction between proteins at each stage of evolution. The probability of each protein-protein interaction is calculated under a proposed stochastic model of network growth and divergence. The topology of the ancestral network (and each intermediate network) is determined by the most probable interactions. Finally, we identify modules in the ancestral network and project them back onto the input interactomes to determine the alignment. We apply the CAPPI procedure to align PPI networks of yeast (*S. cerevisiae*), fly (*D. melanogaster*), and worm (*C. elegans*), which are among the largest interactomes available to date.

We now discuss the details of our approach. The subsequent steps of the analysis are outlined in Fig. 5.1. The original version of the material of this chapter was presented at the ISMB/ECCB 2007 conference and published in Dutkowski & Tiuryn (2007). The sequence comparison and tree building methods used here are described in Chapter 2.

5.1 Reconstructing phylogenetic history

We assume that we are given a number of PPI networks, each coming from a different species. The first step towards aligning the networks is to determine the homology relationship between all proteins. We choose to split the proteins into non-overlapping families of (putatively) homologous proteins (a similar approach was presented in Flannick *et al.*, 2006). The proteins in each cluster are believed to have descended from a common ancestral protein. In contrast to the approach taken by Flannick *et al.* (2006), where proteins are greedily assigned to families during the iterative alignment process (so as to maximize the scoring function), we determine the homology relationships directly, by previously established methods for identification of protein families. To allow the application to arbitrary species on genome-scale, we identify the homologous clusters using the MCL algorithm by Enright *et al.* (2002) with BLAST E-values as pairwise distances between proteins.

In the next step, we reconstruct the evolutionary history of each protein family by means of phylogenetic analysis. To this end, we perform multiple alignment of protein sequences using the CLUSTALW method (Higgins *et al.*, 1994). Next, we calculate the distance matrix using the PROTDIST procedure and construct phylogenetic trees using the neighbor joining algorithm implemented in the Phylip package (Felsenstein, 2005).

For consistency with the evolutionary history, it is necessary to reconcile the gene tree of each protein group with the species tree of the aligned organisms (Page & Charleston, 1997). The reconciliation algorithm minimizing the duplication and loss score function is implemented in NOTUNG (Durand *et al.*, 2006).

Given the sequence of evolutionary events extracted from the reconciled trees and the interactions observed in the input networks, we proceed to determine the posterior probability of interactions between proteins at previous stages of evolution. The reconstructed phylogenies of protein families serve as a backbone for reconstructing protein interactions. Below we present a formal description of the network reconstruction procedure, the proposed model of network evolution, and the details of calculating the posterior probability of ancestral interactions.

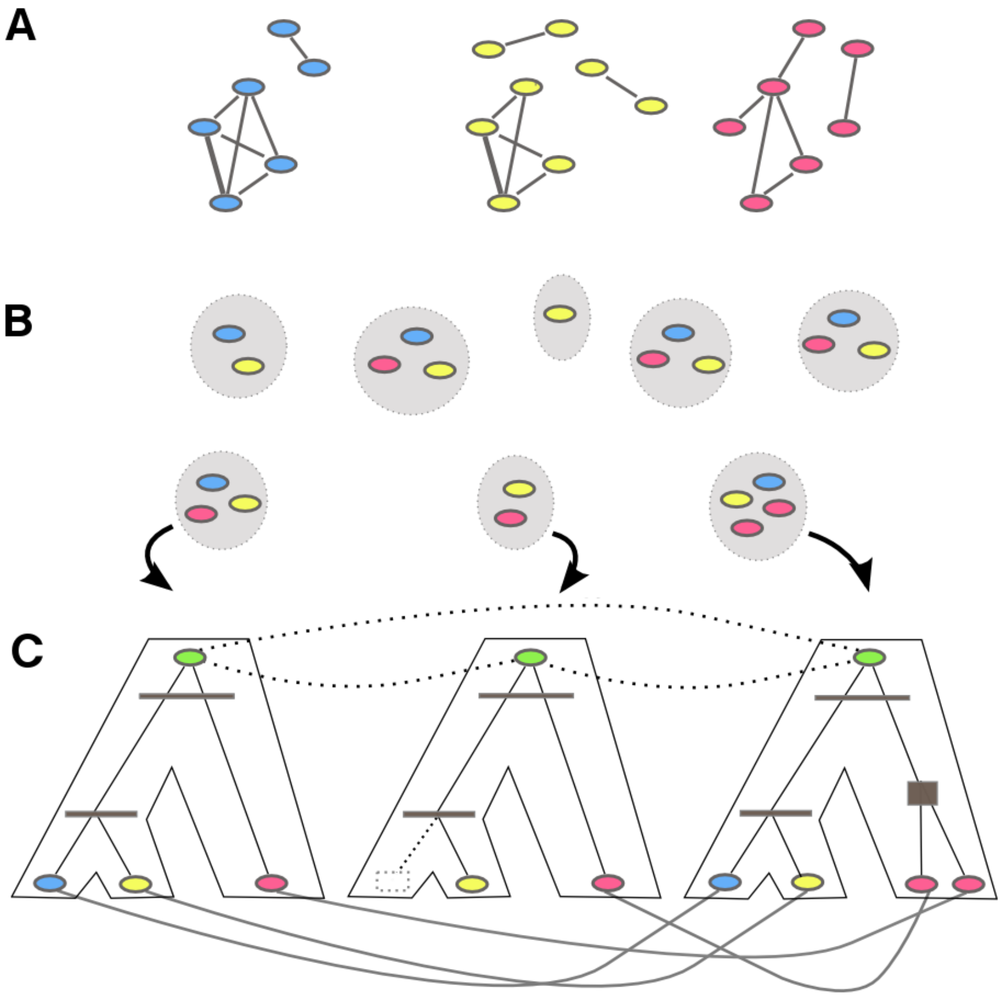


Figure 5.1: *Overview of the analysis performed by CAPPI. Color-coded fragments of three input PPI networks are shown in (A). First, we determine non-overlapping homologous protein groups via sequence clustering (B). Next, we build gene trees for each protein group and reconcile the trees with a common species tree – reconciled trees for three protein groups with outgoing arrows are shown in (C). Finally, we determine the probability of each ancestral protein interaction (dotted lines) given the interactions observed in the input networks (mapped to the leaves of the trees) and the sequence of evolutionary events (duplications, speciations and deletions) from the reconciled gene trees.*

5.2 Reconstructing the ancestral network

Let S_O be the set of species for which we observe input protein-protein interaction networks. We assume that we are given a phylogenetic tree of the considered species in which leaves are labeled with the species from S_O and inner nodes are labeled with unknown (hidden) predecessor species. The tree is determined by the indexed set of nodes $S = \{s_1, s_2, \ldots, s_n\}$ of all the species (observed and hidden) and three functions $LS, RS, F : \{1, \ldots n\} \rightarrow \{null, 1, \ldots n\}$. We assume that the common ancestor (the root of the tree) is s_1. $LS(i)$ and $RS(i)$ return the index of the left and right child of species s_i, respectively (or *null* if the right (left) child does not exist), and $F(i)$ returns the index of the father node (or *null* if $i = 1$). We also assume that the set of proteins from all observed species is split into non-overlapping protein families (equivalence classes) and for each family we are given a gene tree, reconciled with the above-mentioned species tree. From the reconciled gene trees we are able to extract the set of duplication events taking place in species s_i – that is after the speciation event which established s_i and before the speciation event in which s_i evolved into two distinct species. For the purpose of this discussion we assume that we can impose an ordering on duplication events in each species (corresponding to the chronological order). In practice, the reconciled trees provide only a partial order of events – the order between duplications occurring within the same species, but in different branches of the tree or in different trees, is not established. To deal with this problem, we sample the space of possible orderings and average out the results. We have also found, that the particular way that the partial order is extended to a total order has negligible effect on the final outcome of our analysis. Based on this finding, we can speed up the computations by arbitrarily selecting an ordering which agrees with the partial one. A natural strategy is based on the assumption that each path in the reconciled tree between one speciation and the next (corresponding to the evolution of proteins of one species) has the same length. If we further assume that the duplications along each such path in the tree occur in constant time intervals, we can impose a more "balanced" ordering. For simplicity, we also do not discuss the protein deletion events present in the reconciled trees.

Deletions do not complicate the model, but only prune the set of possible protein pairs that have to be considered.

Let $G_{i,j} = (V_{i,j}, E_{i,j})$ denote the graph representing the protein network of species s_i after the j-th duplication event occurring in this species. The ancestral graph is denoted by $G_{1,0}$. Graph $G_{1,0}$ has exactly one protein from each protein family – the protein placed at the root of the appropriate gene tree. The sequence of duplication events in species s_i is given by $D_i = (d_1^{(i)}, \ldots, d_{m_i}^{(i)})$, where $d_j^{(i)} = (n_p, n_a, n_b)$ denotes a duplication of protein $n_p \in V_{i,j-1}$ into two proteins $n_a, n_b \in V_{i,j}$. The problem we consider is to determine the probability of interaction between nodes in the graph $G_{1,0}$ and, similarly, in all its descendants. This is done based on the observed graphs of the species in S_O and assuming the sequences of evolutionary events by which each of the observed networks evolved from $G_{1,0}$.

5.3 Duplication and divergence model of protein network evolution

The following model of network evolution, motivated by the general duplication model (Sole *et al.*, 2002; Pastor-Satorras *et al.*, 2003), is used to determine the probability of observing graph $G_{i,j}$ under the assumption of the sequence of speciations and duplications, by which $G_{i,j}$ evolved from the ancestral network $G_{1,0}$ (see Fig. 5.2 (A) for an example).

The model has four parameters: p_d, δ_d, p_s, and δ_s.

- We start with the ancestral graph $G_{1,0}$, and perform a defined sequence of duplications and speciations. For simplicity, we assume that the initial graph $G_{1,0}$ does not contain self-loops. However, they are easily incorporated into our model and correspond to the evolutionary predecessors of interactions between homologous proteins.

- In case of a **duplication** $d_j^{(i)} = (n_p, n_a, n_b)$ graph $G_{i,j}$ is constructed on the basis of $G_{i,j-1}$ in the following way:

  D1. All vertices besides n_p and edges which are not incident to n_p are copied from $V_{i,j-1}$ to $V_{i,j}$.

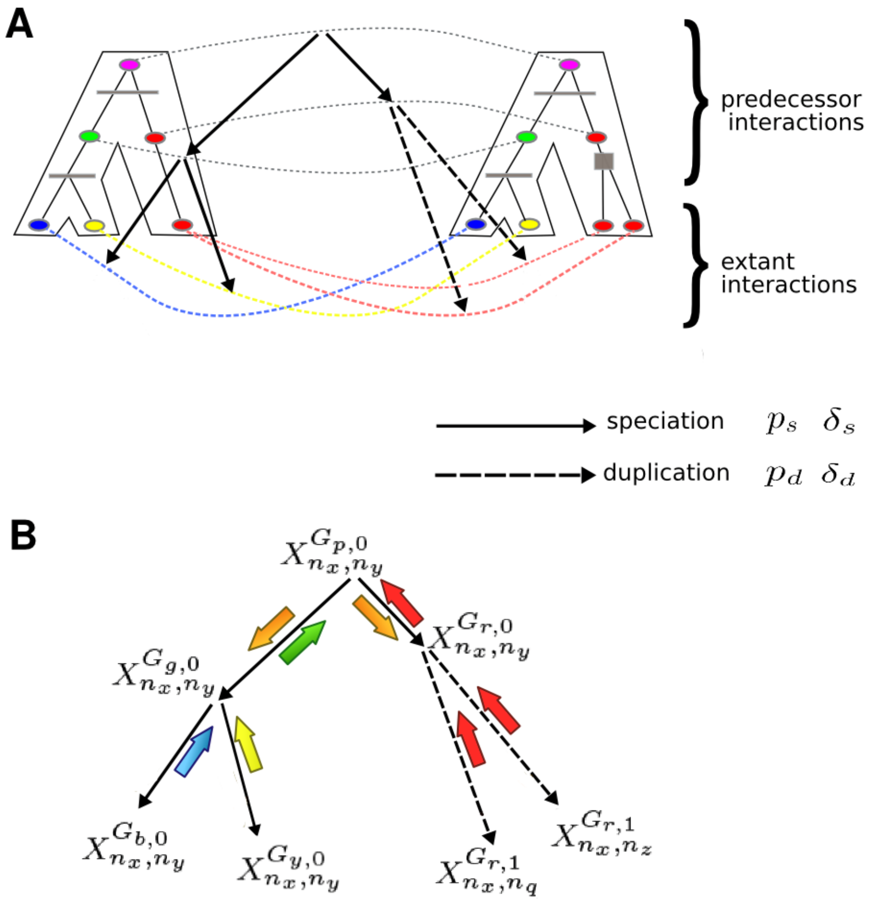


Figure 5.2: *A toy example of the Bayesian tree model of evolution of interactions between members of two protein families for three species: blue, yellow and red. Part (A) shows two reconciled trees for the considered families together with putative protein interactions at each level of evolution. The proteins in the trees are represented by ellipses (colored accordingly to their species). The speciation events are marked by horizontal lines and the duplication events are marked by filled squares. The evolution of the putative ancestral interaction between the root proteins (purple) can be traced down the trees to the extant interactions. In (B) a random variable is associated with each putative interaction. A solid arrow indicates a dependence between two random variables which comes from a speciation event. Similarly, a dashed arrow indicates a dependence for a duplication event. The four parameters (p_s, δ_s, p_d and δ_d) determine the probability of retaining or gaining an interaction in each case. Arrows colored blue, yellow, red and green represent messages corresponding to interaction evidence coming from each of the species. These messages are passed up the tree in the first phase of the MP algorithm. In the second phase, messages containing aggregated evidence from one side of the tree are passed down to the other side (orange arrows).*

D2. Vertices n_a and n_b are added to $V_{i,j}$.

D3. For each edge $n_p n_x \in E_{i,j-1}$ we add to $E_{i,j}$ edges $n_a n_x$ and $n_b n_x$ independently, each with probability p_d.

D4. For each vertex $n_y \in V_{i,j}$ such that $n_p n_y \notin E_{i,j-1}$ we add to $E_{i,j}$ edges $n_a n_y$ and $n_b n_y$ independently, each with probability δ_d.

- In case of **speciation** of species s_i graph $G_{LS(i),0}$ is constructed on the basis of G_{i,m_i} in the following way[1]:

S1. All vertices are copied from V_{i,m_i} to $V_{LS(i),0}$.

S2. Each edge $n_x n_y \in E_{i,m_i}$ is added to $E_{LS(i),0}$ independently with probability p_s.

S3. Each edge $n_x n_y \notin E_{i,m_i}$ is added to $E_{LS(i),0}$ independently with probability δ_s.

Steps D3. and D4. associated with duplication events are referred to as local or correlated divergence because they only effect the edges of the newly added vertices. The steps following speciation can be referred to as global divergence, as they can effect any edge in the network. Wagner (2001) points out that duplicate gene products usually diverge quickly and loose common interactions. One reason for this is that there is greater tolerance for mutations of the newly duplicated proteins because of functional and structural redundancy of the duplicates. This would suggest that p_d (the probability of edge conservation following duplication) should generally be much lower than p_s (the probability of edge conservation after speciation).

Under the proposed model, the probability of interaction between proteins in the network $G_{i,j}$ is determined by the interactions in the ancestral network $G_{1,0}$ and the assumed sequence of speciations and duplications which led to the formation of $G_{i,j}$. In the following, we use the above model to infer the posterior probabilities of ancestral interactions given the observed input networks.

[1] Graph $G_{RS(i),0}$ is constructed independently in the same manner.

5.4 The most probable ancestral interactions

For each graph $G_{i,j}$ and each pair of vertices $n_x, n_y \in V_{i,j}$ we denote by $X^{G_{i,j}}_{n_x,n_y}$ the binary random variable equal 1 when there exists an edge $n_x n_y \in E_{i,j}$, and 0 otherwise.

Assuming the duplication and divergence model described earlier the probability $P(X^{G_{i,j}}_{n_x,n_y} = 1)$ of interaction between between vertices n_x, n_y in the graph $G_{i,j}$ depends on the existence or lack of an edge between the protein pair being the direct predecessor of the pair (n_x, n_y). Let us consider the last evolutionary event which could effect the pair (n_x, n_y). Three cases are possible:

1. Vertex $n_x \in V_{i,j}$ was created by duplication $d^{(i)}_k$ from vertex $n_p \in V_{i,k-1}$, where $k \leq j$ ([1]). Then we have $P(X^{G_{i,j}}_{n_x,n_y} = 1 | X^{G_{i,k-1}}_{n_p,n_y} = 1) = p_d$, and $P(X^{G_{i,j}}_{n_x,n_y} = 1 | X^{G_{i,k-1}}_{n_p,n_y} = 0) = \delta_d$.

2. Vertex $n_y \in V_{i,j}$ duplicated from vertex $n_q \in V_{i,k-1}$, where $k \leq j$ (symmetrical to 1).

3. Vertices $n_x, n_y \in V_{i,j}$ emerged by means of speciation from vertices $n_x, n_y \in V_{F(i),m_{F(i)}}$. We then have $P(X^{G_{i,j}}_{n_x,n_y} = 1 | X^{G_{F(i),m_{F(i)}}}_{n_x,n_y} = 1) = p_s$, and $P(X^{G_{i,j}}_{n_x,n_y} = 1 | X^{G_{F(i),m_{F(i)}}}_{n_x,n_y} = 0) = \delta_s$.

The above dependencies can be represented using a Bayesian network (BN) model (see Fig. 5.2 (B) for an example). We start the construction of the BN from the instantiated random variables which represent the edges or non-edges in the observed graphs of the species in S_O. By considering the last duplication or speciation event, we recursively determine the direct predecessor of each possible edge (random variable) and assign the conditional probabilities, as described above, until we reach the corresponding possible edge in the ancestral graph.

Each random variable corresponding to a possible edge $n_x n_y$ depends on exactly one random variable denoting the edge (or non-edge) in the direct predecessor graph. Therefore the considered BN is a set of trees. Each tree models the

[1] If $k < j$ then there were other duplication events $d^{(i)}_{k+1}, \ldots, d^{(i)}_j$ in species s_i, which did not effect the protein pair (n_x, n_y).

joint distribution of the random variables corresponding to interactions which are descendants of one of the interactions in the ancestral graph.

We can formulate the problem of finding the ancestral graph as a Bayesian inference problem. Precisely, we would like to determine for every i, j, x and y the posterior probability

$$P(X^{G_{i,j}}_{n_x,n_y} = 1|E)$$

of interaction between a protein pair (n_x, n_y) in species i after the j-th duplication, given the set of instantiated variables E, which are the interactions and non-interactions between nodes in the networks of present-day species. We assume that the *prior* probability p_1 of interaction between proteins in the ancestral network $G_{1,0}$ is given. The evolutionary model provides the conditional probabilities linking each child node in the BN to its father. The problem solved here is the classical problem of inference in Bayesian networks. The tree structure of our BN model enables an efficient solution using the message passing (MP) algorithm due to Pearl (1988) (see Chapter 4).

5.5 Identifying conserved ancestral modules

Our ultimate goal is to determine the conserved functional modules in the observed networks. Most of the previously proposed network alignment procedures assumed a specific topology of functional modules. The candidate network regions were scored for fulfilling the desired structure. In contrast to the previous methods, we do not impose a predefined topology when searching for conserved modules. Instead, we identify highly probable connected subnetworks in the ancestral graph, and project them onto extant networks.

The ancestral network $G_{1,0}$, reconstructed according to the procedure described in previous sections, is a complete graph in which each edge $n_x n_y$ is assigned a weight corresponding to the probability of interaction between the adjacent proteins n_x and n_y. The subsets of vertices connected by highly weighted edges are likely to constitute functional modules. To identify the modules of the ancestral network, we set an edge threshold value t and eliminate from the graph $G_{1,0}$ all edges with weights below that threshold. Note that alternatively, we could use a clustering procedure or a search heuristic to identify dense clusters or pathways in the ancestral network.

As shown in the next section, the threshold value can be determined by observing the gradual decomposition of the largest component. At low values of t the nodes of the ancestral network form one giant component. As we raise the value of the threshold t, the giant component decomposes and many components with heavy edges are revealed. The cut-off value can further be refined by determining the level at which the probability of interaction between two vertices in the ancestral graph is statistically significant, compared to the background model. To estimate the level of edge significance, we repeatedly run our algorithm on randomized versions of the input data, maintaining the original homology relationships of the proteins and their phylogenetic history, and permuting the protein interaction data. The original PPIs are randomized by redistributing the edges of the input networks, while maintaining their node degree sequences (a similar technique was applied by Kelley *et al.*, 2003; Koyuturk *et al.*, 2006; Sharan *et al.*, 2005a). For each set of randomized networks we reconstruct the ancestral PPI network and count the number of edges with weights exceeding a given threshold. Next, we compute a q-value based on the false discovery rate (FDR) for multiple hypothesis testing. The FDR q-value for a given edge weight w is estimated as:

$$q(w) = \frac{\sum_{i=i}^{N} C_i(w)/N}{C_{real}(w)},$$

where $C_{real}(w)$ is the number of edges with weights equal to or higher than w in the reconstructed ancestral network and $C_i(w)$ $(i = 1 \dots N)$ is the number of such edges in the i-th randomized network.We determine the threshold value t at which the edge weights are significant (e.g. q-value < 0.05). We then decompose the graph $G_{1,0}$ by deleting the edges with weights lower than t and remove all nodes without any interactions. The remaining connected components of the network constitute the *ancestral network modules.*

With the putative ancestral modules at hand, we proceed to identify the respective descendant modules in each of the considered species. To this end we project the ancestral modules onto the input networks by mapping the nodes of the ancestral network (ancestral proteins) to their descendants (proteins) in the input networks and identify the conserved descendant interactions.

5.6 Experiments and discussion

We apply our method to search for conserved functional modules in the networks of *S. cerevisiae*, *C. elegans* and *D. melanogaster*. In the following we present our results and comparison with previously proposed methods. All algorithms are used with their default parameters, except were noted.

5.6.1 Ancestral network reconstruction

We downloaded PPI and protein sequence data from the Database of Interacting Proteins (DIP) (Salwinski *et al.*, 2004, April 2006 download). In order to identify protein families, we performed MCL clustering (I = 1.2) of all the sequences available in the DIP database (including sequences from species other that the three species of interest), taking BLAST E-values as the pairwise distances. We found that better clustering results are achieved when protein sequences from a larger number of species are included. This is intuitively correct: as homologous proteins should be more similar to each other than to other proteins, a larger and more diverse protein universe serves as a better background for homology identification. The clustering identified 6971 non-overlapping protein groups, 460 of which included protein representatives from all three species (yeast, fly and worm). We decided to limit further analysis to only these 460 clusters. This implies that our protein groups have support in at least 3 species, which is consistent with the major requirement assumed in the construction of orthologous groups in the COG database (Tatusov *et al.*, 2003).

Next, for each of the 460 protein clusters separately, we performed multiple sequence alignment by CLUSTALW, calculated the distance matrix using PROTDIST and constructed a family gene tree using the neighbor joining algorithm. This resulted in an unrooted tree for each family, which was rooted and reconciled with the species tree of yeast, fly and worm using the procedures implemented in NOTUNG.

For each pair of protein families we calculated the posterior probability of interaction between the respective ancestral proteins according to the model described earlier. The choice of model parameters is discussed in the next section. In an analogous way, we computed the probability of each ancestral self loop (representing the interactions of proteins with themselves), based on the observed interactions within one protein family. The resulting ancestral network consisted of 460 nodes and $460 * 459/2 + 460 = 106030$ edges weighted by the probability of interaction of adjacent nodes.

5.6.1.1 Estimating parameters of edge dynamics

Here we discuss the choice of parameters p_d, δ_d, p_s, δ_s and the *prior* probability p_1 of interaction between any two ancestral proteins. The model of network evolution presented in Section 5.3 is motivated by and related to the proteome growth model of Sole *et al.* (2002), often referred to as the general duplication and divergence model. The original model included only the parameters of link conservation or emergence following the duplication event (corresponding to p_d and δ_d in our model). Sole *et al.* (2002) denoted the probability of edge deletion following duplication event by δ and set its value to 0.53, which was influenced by estimations of rates of link addition and deletion made by Wagner (2001). The model of Sole *et al.* (2002) assumes, however, that after a duplication event one of the duplicates remains unchanged. Our model is symmetrical in the sense that both of the duplicates are subject to deletion and emergence of edges. Our first estimate of the probability of edge conservation p_d was therefore 0.7, which (assuming independence) gives almost the same joint probability of edge conservation in both duplicates as in Sole *et al.* (2002). The probability that a new edge is introduced was estimated by Sole *et al.* (2002) as 0.06 divided by the number of nodes in the network. For the purpose of the experiments, we used a constant value $\delta_d = 0.01$, without normalizing it further by the size of the network at each stage. The probability of edge conservation after the speciation event was set to $p_s = 0.95$. This parameter is related to the overall divergence of the network over time and should be more conservative than the probability p_d related to the fast divergence of interactions of newly duplicated proteins. The probability of edge emergence after speciation was set to $\delta_s = 0.01$. We set the probability p_1 of each possible ancestral interaction at 0.01, which is motivated by the observation that PPI networks are generally sparse, with a small average node degree. As shown in the following section, these parameters yielded satisfactory results in terms of the identified modules. We further studied the effect of individual parameters (additional results are available on our web page: http://bioputer.mimuw.edu.pl/papers/cappi) on the reconstructed alignment and chose a more conservative setting, raising the probability of edge conservation and lowering the probability of edge emergence. Precisely we set $p_d = 0.85$, $p_s = 0.99$ and $\delta_d = \delta_s = 0.001$. This allowed us to identify more modules that were well conserved across the considered species.

5.6.2 Decomposition of the giant component

As stated above, the reconstructed ancestral network is a complete weighted graph with 460 nodes. However, the weights of the edges (probabilities of interaction) vary considerably depending on the evolutionary history and the evidence supporting a given interaction in the input networks. As we gradually eliminate the edges with the lowest weights, the initially connected graph decomposes into a large number of small components – suggesting the existence of network modules. This is consistent for a wide range of parameters we have tested. The exact transition point and speed of decomposition varies with the choice of model parameters (especially the choice of the prior probability of interaction); however the general phenomenon is always observed. The decomposition of the ancestral network constructed with the model parameters discussed earlier is presented in Fig. 5.3.

5.6.3 Edge weight threshold selection

The connected components of the ancestral graph, identified at a sufficiently high edge threshold level, determine the ancestral modules of conserved interactions. To determine the significance level we calculated the FDR edge weight q-values using randomized networks. We then deleted all edges with weights below threshold 0.48 (q-value of 0.049). We also eliminated all nodes without any interactions.

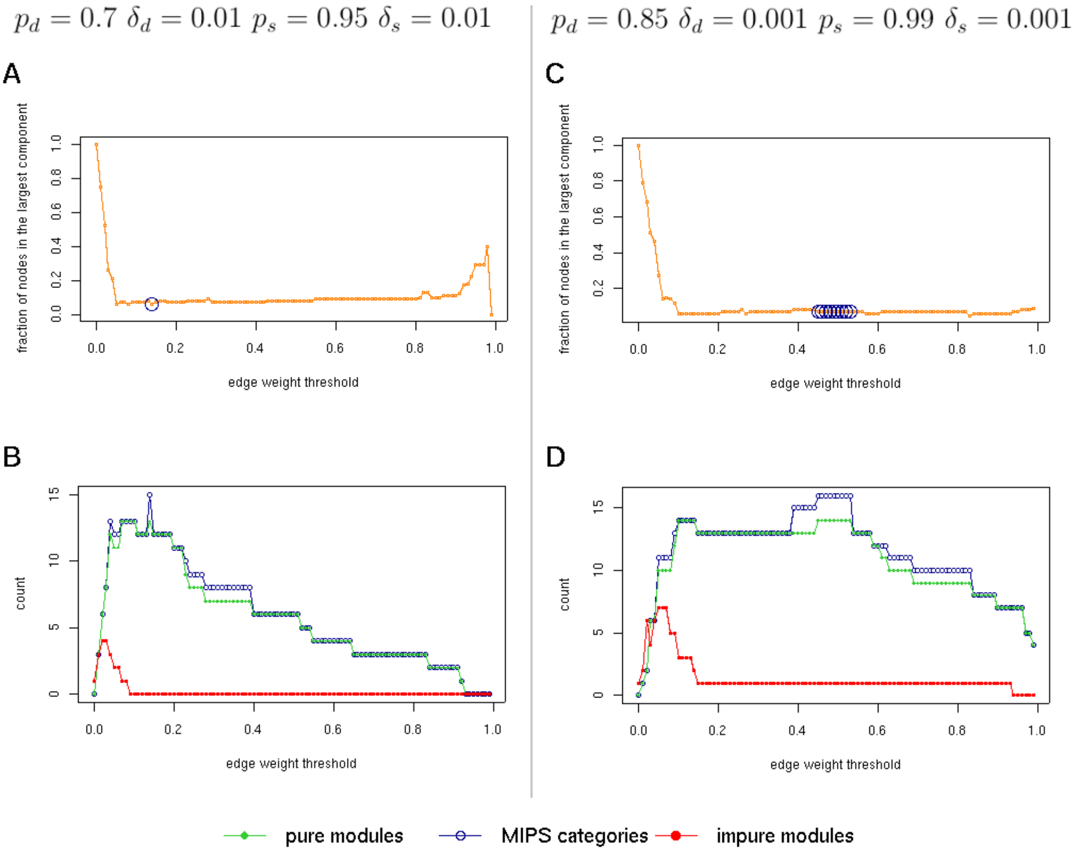


Figure 5.3: *Decomposition of the largest component (A and C) and change in the number of modules and identified MIPS complexes (B and D) with increasing edge threshold. The results for the less conservative parameters are presented in panels A and B, and for the more conservative parameters (preferred) in panels C and D. In case of both settings, as the giant component decomposes, many pure modules and, in consequence, many MIPS complex categories are identified. The nodes of the ancestral network not involved in any interactions are eliminated from the graph, which explains why the fraction of nodes in the largest component (plot A) increases suddenly at high threshold values. The optimal threshold values (in terms of identified pure modules) are indicated in A and C with blue circles.*

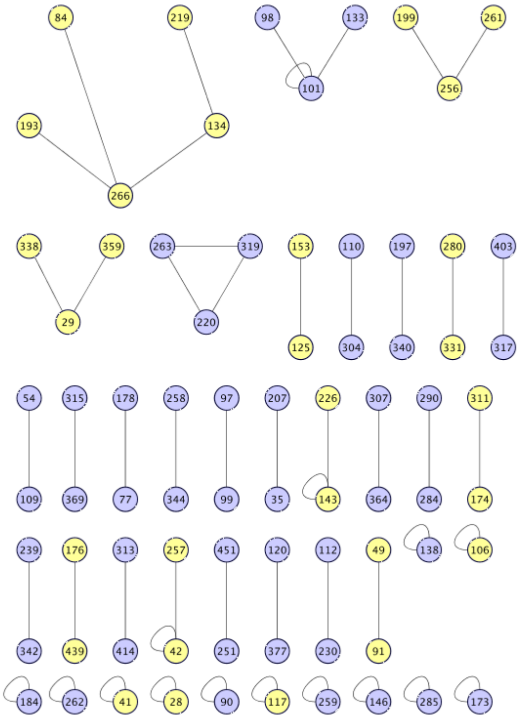


Figure 5.4: *The ancestral network modules identified at edge threshold value of 0.48. Pure modules (colored yellow), matching known protein complexes, are described in Table 5.1. The projection of the largest module (top left) onto the input PPI networks is presented in Fig. 5.5.*

5.6.4 Conserved modules and quality assessment

The ancestral network, decomposed by eliminating edges below the threshold value of 0.48, contains 40 modules (visualized in Fig. 5.4). Overall 75 nodes representing ancestors of conserved protein families are present in the identified modules. By projecting the ancestral nodes onto their present-day descendants, we obtain an alignment consisting of 40 respective network regions (modules) in the three input networks. We have found that a large part of the detected conserved modules match well to the known protein complexes collected in the MIPS database (Mewes *et al.*, 2006). To formally evaluate the quality of this finding we compute the purity score for each identified module as proposed by Sharan *et al.* (2005a). The purity score of a module with respect to a given MIPS category is defined as the number of module's proteins annotated to that category divided by the number of all annotated proteins contained in that module. The module is defined to be pure if it contains at least 3 annotated proteins and at least half of these share the same annotation (purity ≥ 0.5). The module is impure if it contains at least 3 annotated proteins and its purity with respect to any considered MIPS category is less than 0.5. All other modules are treated as not sufficiently annotated (unknown). Following previous studies (Sharan *et al.*, 2005a,b), we only consider the annotations at MIPS level 3 and exclude annotations based on high-throughput experiments (category 550). Overall out of the 40 identified modules 14 are pure and 1 is impure. The 14 pure modules match to 16 MIPS categories summarized in Table 5.1. The largest ancestral module is found to be pure with respect to the MIPS category 360.10.20 (proteasome). This module, projected onto the three PPI networks of the considered species, is presented in Fig. 5.5. The only one impure module we have found is composed of the ancestral nodes 109 and 54. Closer examination of this module yields interesting results. It contains 15 annotated proteins assigned to five level 3 MIPS categories: 260.20.10 (4 proteins), 260.20.30 (4 proteins), 260.20.20 (4 proteins), 260.30.10 (2 proteins), 260.20.99 (1 protein). The module is thus pure with respect to level 2 category 260.20 (intracellular transport) and the interactions between the members of the respective level 3 categories are perhaps not unexpected.

The effect of edge threshold selection (discussed in the previous section) on the identified modules is visualized in Fig. 5.3. The results for the chosen parameter values are presented in the bottom right panel (D). We observe that immediately after the decomposition of the giant component, the number of pure modules increases to 14 and the number of impure modules drops to 1. Pure modules matching known protein complexes are connected to other pure modules by lighter edges, which disappear as the threshold is raised. It can also be seen that by choosing a sufficiently high threshold (0.96) we can eliminate the one impure module from our solution and still identify 7 pure modules. For comparison, we also plot the results for less restrictive parameters, which were perhaps better motivated by previous studies. Similar observations can be made with respect to these results. In fact, in this case, even at small threshold values we do not find any impure modules.

Module	Annotated proteins	Purity	MIPS category	Description
193 - 266 - 134 - - 219 - 84	7	1	360.10.20	Proteasome
28	14	1	360.10.10	Proteasome
257 - 42	6	0.83	410.40.30	Replication
311 - 174	5	1	500.40.10	Translation
331 - 280	3	1	500.20.10	Translation
176 - 439	4	1	510.190.110	Transcription
176 - 439	4	1	510.190.40	Transcription
41	3	0.67	510.190.130	Transcription
199 - 256 - 261	4	0.5	510.70.20	Transcription
199 - 256 - 261	4	0.5	510.190.10	Transcription
153 - 125	4	1	260.50.20	Intracellular transport
117	3	1	260.30.20	Intracellular transport
91 - 49	4	1	440.30.10	RNA processing
338 - 29 - 359	5	0.6	440.30.10	RNA processing
143 - 226	3	1	140.20.20	Cytoskeleton
106	3	0.67	510.180.20	DNA repair
199 - 256 - 261	4	0.5	230.20.20	Histone acetyltransferase

Table 5.1: *MIPS categories matched by pure modules identified by CAPPI. Pure conserved modules are identified by projecting each ancestral module onto the input PPI networks and assigning the annotations from the MIPS database to the respective yeast proteins. The identifier of the ancestral module (see also Fig. 5.4), the number of annotated yeast proteins, and purity with respect to the matching MIPS category is shown. Note that some proteins (members of modules 176-439 and 199-256-261) are annotated to more than one MIPS category.*

5.6.5 Comparison with previous methods

Our approach implemented in the CAPPI framework addresses two main goals. First, it is rooted in an evolutionary network growth model, based on edge dynamics and phylogenetic information about the history of network constituents. Second, it is able to simultaneously align multiple networks. Of the previously

Method	Database	Modules	Pure	Impure	Unknown	MIPS in pure
NetworkBLAST	DIP 2004	183	80	4	99	5
CAPPI $q\text{-value} < 0.05$	DIP 2004	39	10	1	28	13
CAPPI $q\text{-value} < 0.03$	DIP 2004	22	7	0	15	7
CAPPI $q\text{-value} < 0.05$	DIP 2006	40	14	1	25	16
CAPPI $q\text{-value} < 0.03$	DIP 2006	22	7	0	15	7

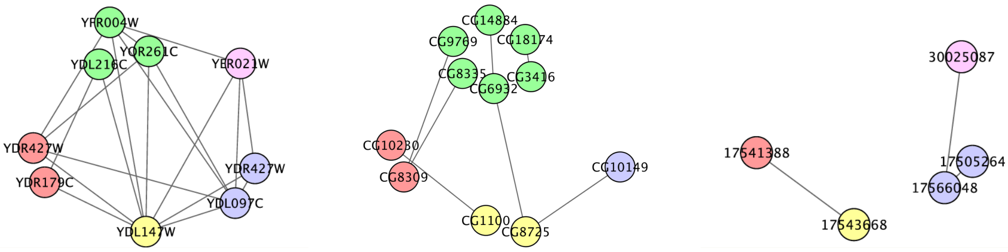
Table 5.2: *Comparison of CAPPI and NetworkBLAST results. With the original settings NetworkBLAST returns more modules altogether and a higher fraction of pure modules. In contrast to NetworkBLAST, CAPPI returns non-overlapping results. Thus less modules are returned, but among them more distinctive MIPS categories and less impure modules are identified.*

available network alignment methods, only NetworkBLAST (Sharan *et al.*, 2005b) and Graemlin (Flannick *et al.*, 2006) have demonstrated the ability to align multiple (> 2) networks. Graemlin's implementation is publicly available, however, the algorithm has many network dependent parameters which have only been estimated for the SRINI networks used by the authors. The estimation of these parameters for other networks was not supported by the original implementation. NetworkBLAST was previously used to align the DIP networks of yeast, worm and fly (Sharan *et al.*, 2005b) and the results provided by the authors enabled a straightforward preliminary comparison of the methods. NetworkBLAST, being one of the first methods developed, became a benchmark used by many authors of new network aligners. Thus comparing against it may provide a relative assessment of whether our algorithm is competitive to other methods in the field. Below we compare the results of the two algorithms using the MIPS complex database as a reference set of true functional modules.

NetworkBLAST experiments were performed on an earlier version of the DIP database (February 2004), which contained fewer interactions and protein sequences. In order to allow a fair comparison, we have repeated our experiments on the same version of the database. The results are summarized in Table 5.2. Overall NetworkBLAST returned a much larger number of aligned modules. Also, a larger fraction of the returned modules were pure. However, the identified modules overlap considerably (up to 80% overlap was allowed by the authors) and among the 80 pure modules only 5 different MIPS categories are matched. In contrast, the alignments returned by CAPPI are non-overlapping. We provide CAPPI results for two edge weight thresholds. With a less restrictive edge threshold (corresponding to q-value < 0.05) our method returned 39 modules among which 10 were pure. The 10 pure modules matched 13 MIPS complex categories altogether. The only one impure module returned was the same as the impure module identified in case of the 2006 DIP version described earlier.

Raising the edge threshold to a more stringent level (corresponding to q-value < 0.03), resulted in a smaller number of pure modules and less identified MIPS categories. However, with this setting no impure modules were returned by CAPPI. At the same time CAPPI still identified two more MIPS categories than NetworkBLAST. We note that in case of both methods a considerable number of identified modules contain less than 3 annotated proteins (thus they are not classified as being pure or impure). These modules may represent yet unknown protein complexes. Note that we intentionally do not formally define a sensitivity measure here because the total number of conserved complexes is unknown. Instead, we rely on the absolute number of matched MIPS complexes to compare the ability of the methods to identify true complexes. The assessment of the specificity of both methods is based on the number of impure modules divided by the number of all identified modules and subtracted from one. With the more restrictive edge threshold CAPPI achieves a perfect specificity score.

Overall, although not exhaustive, the presented comparison demonstrates CAPPI's competitiveness and provides evidence for its relatively high sensitivity and specificity. Interestingly, an independent study by Chagoyen *et al.* (2008) has recently shown that practically all modules identified by CAPPI are functionally coherent and statistically significant with respect to biological process annotations from Gene Ontology (Ashburner *et al.*, 2000). CAPPI was the only network alignment method evaluated, and it achieved the highest specificity, at the cost of identifying fewer modules than methods searching for coherent subgraphs only in the yeast network. Other scoring frameworks can be applied to



S. cerevisiae *D. melanogaster* *C.elegans*

Figure 5.5: *The conserved modules identified by projecting ancestral module 193-266-134-219-84 (see also Fig. 5.4) onto the networks of yeast, fly and worm. Seven of the nine yeast proteins are assigned to the MIPS category 360.10.20 (proteasome). The families corresponding to ancestral nodes 193, 266, 134, 219 and 84 are colored red, yellow, blue, purple and green respectively. All 3 species contain representatives of each protein family, but only interacting proteins are visualized to present truly conserved PPI regions. The overall small coverage of C. elegans protein interactions in DIP may explain the apparent weak conservation of this module in case of this species.*

further assess the performance of our approach. We will explore them in a different context in the following chapters.

For completeness we also provide the results for the DIP 2006 version discussed in the previous section. We observe that at the lower edge threshold (corresponding to q-value < 0.05) CAPPI identifies more pure modules and MIPS categories on the more complete 2006 DIP database than on the 2004 edition. The results obtained with the more restrictive threshold remain unchanged with the introduction of new proteins and interactions, indicating that the group of alignments with the best support in the data is the same for both database versions.

We note that following the publication of our analysis (Dutkowski & Tiuryn, 2007), a similar model was independently used by Pinney *et al.* (2007) to recover ancestral states of PPIs within a single family of the bZIP transcription factors.

Chapter 6

Phylogeny-guided interaction mapping in seven eukaryotes

In this chapter we extend our modeling framework and apply it to a different task. We develop a comprehensive method for integrating PPI evidence from different datasets and transferring it across species. It is designed towards three basic goals. First, interactomes from different species should be compared and integrated in the context of their evolution. Therefore the processes by which protein interaction networks grow and diverge over time should be accounted for in the framework. Second, when predicting the interactions of a given protein we should take into account the PPI evidence from all similar proteins (homologs), because the role of an individual protein in one species may be distributed over several proteins in another species. This strategy is also motivated by the scarceness of the source datasets from which new interactions can be inferred. Thus it is better to take advantage of all relevant information. Third, the impact of each data source on our final result should be based on its inherent reliability and coverage. A dataset from a small scale study is certain not to contain the vast majority of interactions. On the other hand the interactions it contains are often more reliable than interactions from large high-throughput screens.

Our framework computes the probability of interaction between two proteins by considering all evidence for interaction between members of the respective protein families to which the proteins belong to. The evidence is accounted for in the context of the families' phylogenetic trees and under the described model of network evolution, which assigns probability scores to events of interaction loss or gain, following a duplication or a speciation event. Intuitively, the closer a given pair of proteins is to another pair, the more impact the evidence for one pair has on predicting the interaction of the other pair. Our Bayesian model naturally takes into account the inherent reliability and coverage of each input dataset. The amount and reliability of the evidence, as well as the evolutionary proximity of the observed interactions to the pair of proteins in question, determines the posterior probability of interaction computed by our framework.

Our approach combines and extends the concepts of interlog mapping and Bayesian data integration. First, as opposed to the interlog approach, we employ information from all homologs in each family (relative to their proximity in the tree), instead of using only the single best ortholog for each protein. Second, we use a Bayesian modeling framework to integrate PPI evidence from many experimental sources, taking into account their reliabilities and coverage. As opposed to the Naive Bayes approach for data integration (which assumes independence of data sources), our approach computes the posterior probability of interaction for every pair of proteins under an established duplication and divergence model of network evolution.

We use our framework to integrate and infer new PPIs in seven eukaryotes: *H. sapiens*, *M. musculus*, *R. norvegicus*, *D. melanogaster*, *C. elegans*, *S. cerevisiae*, and *A. thaliana*. We perform a comprehensive validation of our predictions using two independent scoring schemes: GO-based functional similarity and an assessment based on reference datasets of binary and co-complex PPIs. The obtained results demonstrate the ability of our method to identify a large percentage of known interactions in a blind test and provide new hypothesis for experimental verification when all known data is integrated. We show that CAPPI performs better than two previous approaches which map interactions across species. We also analyze specific examples of valid PPI predictions in well-characterized complexes in yeast and human (proteasome, endosome and exosome), and show that core subcomplexes can be accurately recovered based solely on the data from the other species (i.e. without any use of the experimental data from the species of interest). Many of the between-module interactions (possibly species-specific) are harder to transfer from distant organisms. Finally, based on our predictions, we present hypothesis on new proteins interacting with the putative SWI/SNF chromatin remodeling complex in *A. thaliana*. Our results are freely available at http://bioputer.mimuw.edu.pl/cappi.

The material of this chapter was presented at the 2009 Systems Biology: Networks conference at Cold Spring Harbor and was submitted for publication.

6.1 Methods

6.1.1 Integrating diverse experimental data

The model presented in Chapter 5 captures the basic notions of protein network evolution. We previously assumed that the PPI data is free of error and complete and we used the model to make inferences about the ancestral interaction networks. However, due to experimental errors and incomplete sampling, the real interactions and non-interacting protein pairs are not certain. This implies that the experimental data should only be used as supporting evidence of putative interactions. To model this accurately in our framework we keep the random variables corresponding to extant interactions unknown and add another level of random variables corresponding to experimental evidence (see Fig. 6.1 (A)). The evidence in each experimental dataset is weighted by the dataset's reliability.

Let $G_{i,m_i} = (V_{i,m_i}, E_{i,m_i})$ be the extant protein interaction network of a present-day species s_i (we assume that m_i is the final duplication occurring in s_i). Let $O_i = \{o_1^{(i)}, \ldots, o_{k_i}^{(i)}\}$ be the set of experimental datasets for species s_i, where each $o_h^{(i)}$ is the set of protein pairs confirmed to interact in the h-th experiment. Let $Rel(o_h^{(i)})$ be the fraction of elements in $o_h^{(i)}$ believed to be true positives. Let $E'_{i,m_i} = \{(n_x, n_y) : n_x, n_y \in V_{i,m_i} \wedge (n_x, n_y) \notin E_{i,m_i}\}$ be the set of non-interacting protein pairs in the graph G_{i,m_i}. For each experimental dataset $o_h^{(i)}$ we denote by $X_{n_x,n_y}^{o_h^{(i)}}$ a random variable which takes value 1 if interaction (n_x, n_y) is present in this dataset and 0 otherwise. For each pair of proteins (n_x, n_y) and each dataset $o_h^{(i)}$, we set the probability of observing a true interaction to be equal the true positive rate of the experiment, and the probability of observing a false positive

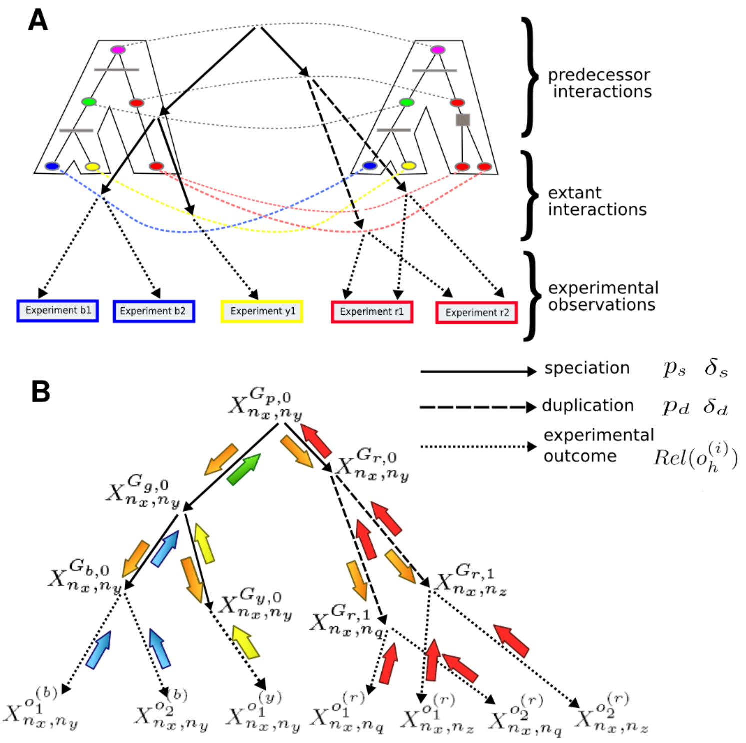

A
predecessor interactions
extant interactions
experimental observations
Experiment b1
Experiment b2
Experiment y1
Experiment r1
Experiment r2
B
speciation p_s δ_s
duplication p_d δ_d
experimental outcome $Rel(o_h^{(i)})$
$X^{G_{p,0}}_{n_x,n_y}$
$X^{G_{r,0}}_{n_x,n_y}$
$X^{G_{g,0}}_{n_x,n_y}$
$X^{G_{b,0}}_{n_x,n_y}$
$X^{G_{y,0}}_{n_x,n_y}$
$X^{G_{r,1}}_{n_x,n_z}$
$X^{G_{r,1}}_{n_x,n_q}$
$X^{o_1^{(b)}}_{n_x,n_y}$
$X^{o_2^{(b)}}_{n_x,n_y}$
$X^{o_1^{(y)}}_{n_x,n_y}$
$X^{o_1^{(r)}}_{n_x,n_q}$
$X^{o_1^{(r)}}_{n_x,n_z}$
$X^{o_2^{(r)}}_{n_x,n_q}$
$X^{o_2^{(r)}}_{n_x,n_z}$

Figure 6.1: *A toy example of the extended Bayesian tree model of evolution of interactions between members of two protein families for three species: blue, yellow and red. This time for each species a certain number of experimental datasets is given: two for blue and red and one for yellow. Part (A) shows two reconciled trees for the considered families together with putative protein interactions at each level of evolution. The evolution of the ancestral interaction between the root proteins (purple) can be traced down the trees to the extant interactions. Evidence for the extant interactions can be found in the experimental datasets. In (B) a random variable is associated with each putative interaction. A solid arrow indicates a dependence between two random variables which comes from a speciation event. Similarly, a dashed arrow indicates a dependence for a duplication event. Finally, dotted arrows represent an interface between the true interactions in extant species and the observed experimental evidence. The parameters p_s, δ_s, p_d and δ_d determine the probability of retaining or gaining an interaction during evolution, while the reliability of each dataset ($Rel(o_h^{(i)})$) determines the probability of identifying a true interaction or a false positive one. As before, arrows colored blue, yellow, red and green represent messages, corresponding to interaction evidence, which are passed up the tree in the first phase of the MP algorithm. In the second phase, messages containing aggregated evidence from one side of the tree are passed down to the other side (orange arrows).*

interaction equal the false positive rate of the experiment, as follows:

$$Pr(X_{n_x,n_y}^{o_h^{(i)}} = 1 | X_{n_x,n_y}^{G_{i,m_i}} = 1) = \frac{Rel(o_h^{(i)})|o_h^{(i)}|}{|E_{i,m_i}|}$$
$$Pr(X_{n_x,n_y}^{o_h^{(i)}} = 1 | X_{n_x,n_y}^{G_{i,m_i}} = 0) = \frac{(1 - Rel(o_h^{(i)}))|o_h^{(i)}|}{|E'_{i,m_i}|},$$

where by $|A|$ we denote the number of elements in the set A. Now each experimentally observed interaction can be naturally incorporated into the BN framework. Similarly each pair not observed to interact in the considered experiment ($(n_x, n_y) \notin o_h^{(i)}$) can be incorporated into the model with conditional probabilities corresponding to the false negative rate and true negative rate of the experiment (see Appendix A for details). The model can also be easily generalized to incorporate distinct reliability values for each single interaction.

6.1.2 Inferring extant protein interactions *via* message passing

The integrated BN model, comprising all PPI edges from every level of evolution and from the experimental datasets, is used to infer protein interactions in the input species. Each random variable corresponding either to a possible interaction, or to a single experiment outcome, depends on exactly one random variable which denotes an edge (or non-edge) in the direct evolutionary predecessor in the first case, and in the network of an extant species in the second case. The considered BN model is a set of Bayesian trees, where each tree represents the joint distribution of the random variables corresponding to putative interactions (which descended from a single edge in the ancestral graph) and the associated experimental evidence (an example of such tree is shown in Fig. 6.1 (B)). As in Chapter 5, the tree structure allows us to apply Pearl's message passing (MP) algorithm to compute the exact posterior probability of interaction between proteins in extant species, in time linear to the number of random variables (see Fig. 6.1 (B) for an example and Chapter 4 for details). Specifically we determine the posterior probability of interaction $P(X_{n_x,n_y}^{G_{i,m_i}} = 1|O)$ for each pair of nodes (n_x, n_y) in each extant network G_{i,m_i}, where O denotes all experimental datasets for all species.

6.2 Experimental setup

We apply CAPPI to infer protein-protein interactions in seven eukaryotic species: human (*H. sapiens*), mouse (*M. musculus*), rat (*R. norvegicus*), worm (*C. elegans*), fly (*D. melanogaster*), yeast (*S. cerevisiae*), and thale cress (*A. thaliana*). The initial steps of our analysis preprocess the data and gather experimental evidence for interaction between members of distinct protein families. To this end, we identify groups of homologous proteins by clustering all non-redundant protein sequences downloaded from the Integr8 database (Kersey *et al.*, 2005) and pull relevant PPI data from IntAct (Hermjakob *et al.*, 2004), MINT (Chatr-aryamontri *et al.*, 2007) and DIP (Salwinski *et al.*, 2004) databases (see Appendix A for details). The family-oriented view of the overlap of available PPI evidence for four best-represented interactomes is shown in Fig. 6.2.

We consider two modes of application of our framework. First, the integration mode which gathers all available input data to provide a reconciled interactome view for each species. Second, the prediction mode which predicts the interactions for each species based only on the evidence from the other species (blind test). To demonstrate the different aspects of our method and enable a straightforward comparison to the previous approaches, we use different combinations of the input datasets and different reliability values (see also Appendix A), yielding the following sets of inferred interactions:

CAPPI-Integ: interactions for all seven species inferred using all available experimental datasets.

CAPPI-Integ-3sp: yeast, fly and worm interactions inferred based on experimental datasets of Ito *et al.* (2001), Uetz *et al.* (2000), Giot *et al.* (2003) and Li *et al.* (2004), with reliability parameters set according to Liu *et al.* (2005).

CAPPI-Pred: interactions inferred for each species using experimental datasets only from the other six species.

We compare the results of CAPPI with the following methods:

Domain-ML: a maximum likelihood domain-oriented method by Liu *et al.* (2005). Yeast interaction predictions, based on experimental datasets of Ito, Uetz, Giot and Li, were provided by the authors.

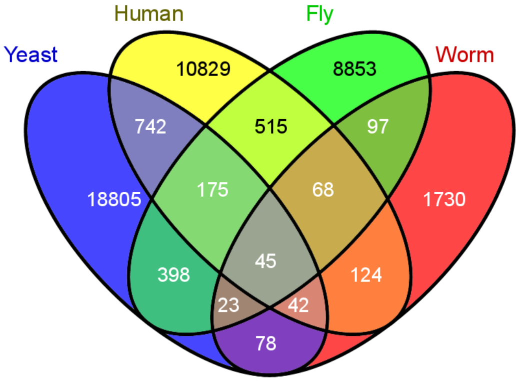


Figure 6.2: *A 4-way Venn diagram illustrating the overlap of PPI evidence between four of the considered seven species: human, yeast, fly and worm. Each cell in the diagram is labeled with the number of pairs of protein families for which members interact in the corresponding species. For example, there are 742 pairs of protein families such that in both yeast and human there exists at least one interaction between members of the two families and no such interactions exist for fly and worm. Only about 0.5% (42514 of 8280415) of possible family pairs we consider have any evidence for interaction in any of the four species. Of these only 0.1% (45 of 42514) have evidence in all four species, which seems small, given that all considered families are evolutionarily conserved. However, the size of the overlap presumably corresponds to the fraction of the interactomes sampled experimentally, rather than to the actual level of conservation. For example, while there is a significant size difference between the overlap of the relatively best sampled yeast and human interactomes (742+175+45+42 = 1004 family pairs) and the overlap between yeast and worm interactomes (23+45+42+78 = 188 family pairs), the fraction of family pairs with PPI evidence from human and worm overlapping with such pairs in yeast is of the same magnitude (8% and 9%, respectively). It is highly probable that many of the homologous interactions in yeast and human have, yet unidentified, counterparts in worm and similarly in the other species.*

Interlog: an interlog-based method implemented by Michaut *et al.* (2008). The program was downloaded from the InteroPorc website `http://biodev.extra.cea.fr/interoporc/Default.aspx` and ran for each species using experimental datasets only from the other six species (same datasets as in CAPPI-Pred).

In the following, we investigate the performance of our method on large-scale data, as well as in small-scale experiments, focused on specific functional modules. We start with a brief description of the quality assessment procedures applied in each case.

6.2.1 Assessing PPI predictions in large-scale studies

In general, the assessment of PPI predictions posses problems due to the limited number of 'gold standard' interactions and the lack of negative test cases. Motivated by previous studies, we employ two scoring schemes to assess the quality of predicted PPIs, as well as those from the input datasets. The first one compares Gene Ontology (GO) annotations (Ashburner *et al.*, 2000) of adjacent gene products and measures their functional similarity. Functional similarity is used as an indirect measure of interaction: the more similar the annotations of the two proteins are, the more confident we are in deeming an interaction between them. We apply a recent information content method of Schlicker *et al.* (2006), implemented in the SemSim R package by Xiao Gou: `http://www.bioconductor.org/packages/2.0/bioc/html/SemSim.html`, which extends the measures previously proposed by Resnik (1995) and Lin (1998). For each pair of proteins we individually measure the similarity of annotations in each of the three ontologies: biological process (BP), molecular function (MF) and cellular component (CC). This results in a *BP* score, *MF* score and *CC* score, respectively, each ranging from 0 (no similarity) to 1 (maximum similarity). When the context allows, we refer to each of these scores as a *GO* score of a pair of proteins.

Our second kind of quality assessment is based on a comparison with a reference dataset. We estimate the ratio of true positive interactions (predictions which are confirmed in a reference dataset) and false positive interactions (unconfirmed predictions for which the two proteins have disjoint cellular localizations).

A similar procedure was applied by Jansen *et al.* (2003). We use separate reference datasets for binary PPIs (direct physical interactions) and for co-complex PPIs (pairs of proteins co-occurring within the same complex). For details on the reference datasets and the localization data see Appendix A. Note that the proper sensitivity and specificity measures are hard to estimate because the reference sets of positive interactions and negative protein pairs are not comprehensive. Due to interdependencies between interactions, implied by our model, cross-validation cannot be easily applied. Instead, in the second part of the analysis, we perform a blind test in which we leave out the data of one species and predict its interactions only based on the data from the other species.

6.2.2 Assessing predictions in functional module case-studies

For small-scale functional module case studies, presented further in this chapter, we report all interactions predicted among a determined set of proteins for a selected threshold value. To assess the statistical significance of interaction predictions, we compute a p-value based on the cumulative distribution function of the hypergeometric distribution, where confirmed interactions are regarded as successes and unconfirmed interactions are regarded as failures. As the predictions are made by CAPPI-Pred which is trained without the use of the input datasets for the predicted species, we use the held out input data as a reference. Note that it is possible that some of the reference interactions are in fact false-positives – an inherent risk of using high-throughput data. In this particular test, however, we are interested in assessing the possibility to predict a significant portion of known PPIs (of which many are from high-throughput studies) by a mapping from other organisms. The reference set is further extended in each case by PPIs curated from specific publications characterizing interactions within the studied complexes. These are as follows: Cagney *et al.* (2001) and Chen *et al.* (2008) for the 26S proteasome PPIs; Hurley & Emr (2006) and Shim *et al.* (2008) for the endosome-related PPIs; Lehner & Sanderson (2004) for the exosome-related PPIs; Sarnowski *et al.* (2002), Farrona *et al.* (2004), Sarnowski *et al.* (2005), Hurtado *et al.* (2006) and Bezhani *et al.* (2007) for the SWI/SNF-related PPIs. Note that for *A. thaliana* there are no high-throughput datasets

Species	CAPPI-Integ			CAPPI-Integ-3sp		
	Data Size	Input Score	Output Score	Data Size	Input Score	Output Score
Yeast	28590	0.377	0.412	1890	0.320	0.381
Fly	12107	0.295	0.425	4049	0.255	0.303
Worm	2604	0.364	0.469	856	0.374	0.485
Arabidopsis	1349	0.596	0.623	NA		
Rat	1271	0.296	0.384	NA		
Mouse	2456	0.417	0.463	NA		
Human	17672	0.353	0.395	NA		

Table 6.1: *Improvement in BP scores over the input datasets. For both CAPPI versions (CAPPI-Integ and CAPPI-Integ-3sp) the number of interactions in each species and the mean BP scores for the input dataset and for the inferred CAPPI dataset of the same size are given. In all cases the inferred interaction set receives a significantly higher score than its input counterpart.*

available, so all reference data for this species come from small-scale studies.

6.3 Integration of interactions in seven eukaryotes

CAPPI-Integ provides an integrated and reconciled view of seven eukaryotic interactomes. Our ultimate goal is to provide a higher quality interactome for each input species. To assess whether this is the case, we perform two separate evaluations using the GO-based scoring scheme and gold standard reference datasets.

6.3.1 GO-based scoring

We first consider the biological process (BP) annotations and score our predictions, as well as the interactions from the input datasets, using the functional similarity measure from Schlicker *et al.* (2006). Mean *BP* scores for the input datasets and for the equal in size prediction datasets are summarized in Table 6.1.

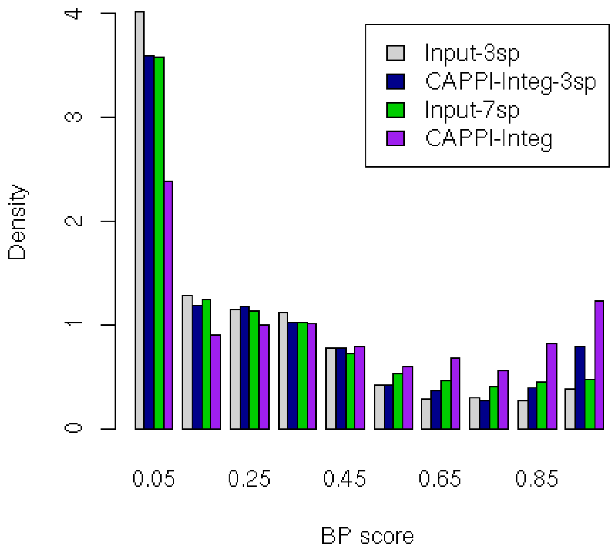


Figure 6.3: *Histogram of BP scores for the fly input datasets (combined) and the corresponding inferred datasets of the same size (4049 PPIs in case of Input-3sp and CAPPI-Integ-3sp, and 12107 PPIs in case of Input-7sp and CAPPI-Integ). Both CAPPI-Integ and CAPPI-Integ-3sp provide higher-scoring interactomes compared to their input datasets, demonstrating the method's ability to increase interactome quality by integrating data from other species.*

We do not consider the scores of self interactions (present both in the input and in the inferred datasets) as they could introduce bias to the results (the GO annotations are identical in this case). Also, to avoid possible bias caused by the specific choice of proteins, input datasets were limited to interactions between members of conserved protein families used by CAPPI (see Appendix A). For each CAPPI version in Table 6.1 we indicate the mean *BP* scores for the input dataset and the inferred output dataset of equal size. For example, in case of CAPPI-Integ the input yeast dataset contains 28590 interactions, for which the average *BP* score is 0.377. The corresponding CAPPI-Integ score of 0.412 was computed by taking the mean *BP* score of the 28590 best predictions in yeast (i.e. interactions with the highest probability). For each of the species CAPPI predictions receive significantly higher mean *BP* scores than the datasets used for training. The most significant improvement over the input datasets is achieved in case of the fly, worm and rat predictions. The mean *BP* score for the entire fly

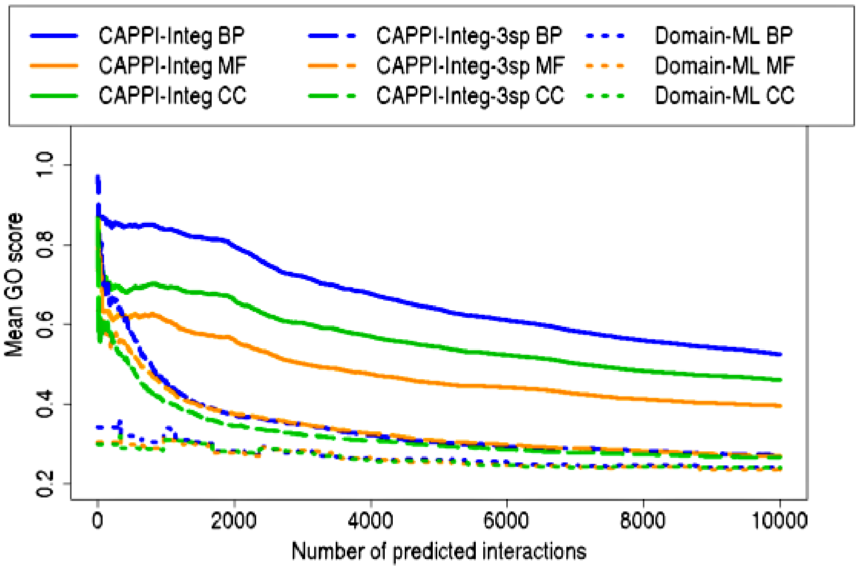


Figure 6.4: *Assessment of inferred yeast interactions using three GO scores. The similarity of GO annotations of each pair of interacting proteins is measured in each ontology: biological process (BP), molecular function (MF) and cellular component (MF). CAPPI and Domain-ML predictions are ranked by their probabilities and the average GO score for a given number of top predictions is shown. CAPPI-Integ-3sp outperforms Domain-ML trained on the same experimental data. CAPPI-Integ integrates all available data from the seven species and further improves the prediction score.*

input dataset is 0.295, while the CAPPI-Integ dataset of the same size achieves a mean scores of 0.425 (44% higher). In case of worm and rat prediction we observe a 29% and 30% increase in the *BP* score, respectively. These results show that CAPPI is able to produce reconciled interactomes which score significantly higher than the input interactomes. A detailed view of the distributions of *BP* scores for experimental and predicted datasets of protein interactions in *D. melanogaster* is presented in Fig. 6.3. The predicted datasets (both CAPPI-Integ and CAPPI-Integ-3sp) contain a lot more high-scoring interactions than the input datasets. Interestingly, while the Input-3sp dataset for fly is almost as good as the Input-7sp dataset, CAPPI-Integ-3sp is significantly outperformed by CAPPI-Integ. This is largely due to the integration of additional high quality datasets from other species, from which CAPPI-Integ can transfer new evidence when inferring the fly interactome.

The improvement in mean *BP* score described above is achieved for relatively large predicted datasets (as large as the initial inputs). As we show in Fig. 6.4, *BP* scores are actually higher for our top predictions. Figure 6.4 plots mean similarity scores according to all tree ontologies: biological process (BP), molecular function (MF) and cellular component (CC), as functions of the number of predicted interactions. The mean scores for both CAPPI versions are negatively correlated with the size of the output dataset. This enables the user to trade size for quality, obtaining a smaller dataset, but of greater reliability.

6.3.2 Testing against gold standard datasets

We further survey the performance of our method using a set of gold standard binary PPIs pulled from (Reguly *et al.*, 2006) and (Yu *et al.*, 2008), as well as co-complex data from the MIPS (Mewes *et al.*, 2006) and CYC2008 (Pu *et al.*, 2009) complex catalogues (see Appendix A for details). Once again, we score CAPPI predictions and compare them to the scores of the input datasets.

The results are presented in Fig. 6.5. The figure plots the ratio of true positive and false positive interactions present among a subset of a given size. The true positive interactions are either confirmed by binary PPIs or known to participate in a characterized complex. The false positives are pairs of proteins with different subcellular localization and thus their interaction is unlikely. Note that in general true interactions constitute only a very small fraction of all possible protein pairs – at most 0.5% in yeast based on recent estimates by Hart *et al.* (2006). This is reflected in our reference datasets. The positive reference used in this case contains 22480 PPIs and co-complex pairs while the negative set contains 4857065 differencially localized pairs (see also Appendix A). It is unlikely to identify a true interaction by pure chance alone. Results presented in Fig. 6.5 confirm the previous observation that reliable interactions are generally ranked high by our method. It is comforting that both CAPPI datasets contain more confirmed interactions than differentially localized pairs among the top ranked predictions (TP/FP >> 1). CAPPI-Integ-3sp has a much higher TP/FP ratio than the input yeast datasets (Ito and Uetz) used for its training. CAPPI-Integ integrates four more high-throughput yeast datasets and consistently scores higher than three

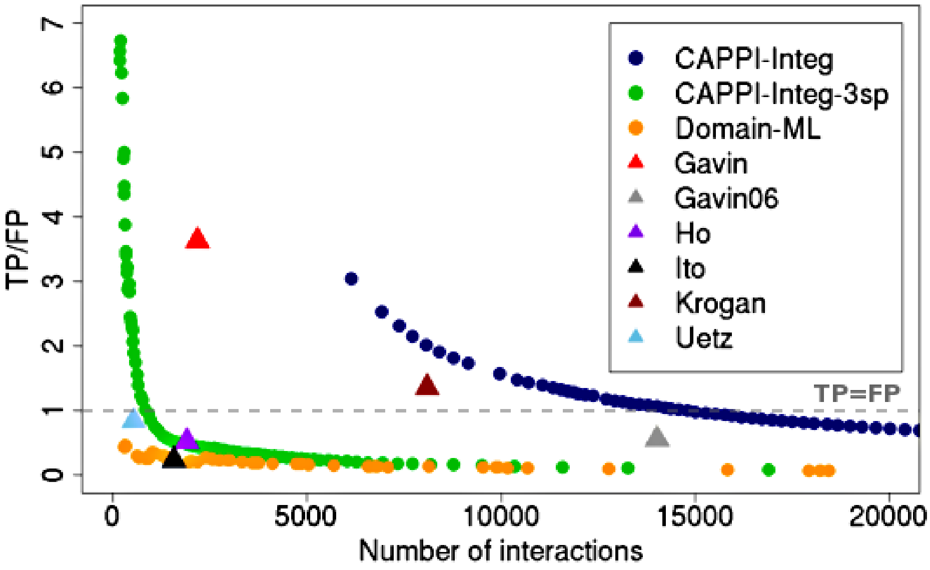


Figure 6.5: *The ratio of true positives (TP) and false positives (FP) as a function of the number of yeast interactions in the CAPPI-Integ dataset. An interaction is deemed true positive, if it is found in the reference dataset comprising co-complex and binary PPIs, and false positive, if the two proteins are assigned different localizations in the MIPS sub-cellular localization catalog. The TP/FP ratios for the CAPPI-Integ, CAPPI-Integ-3sp and Domain-ML predictions are compared with the scores of the input experimental datasets. The gray dashed line marks the level at which the number of true positive predictions is equal to the number of false positive predictions.*

out of four of these inputs – Gavin (2002) dataset has a higher score, but for a smaller number of interactions.

6.4 Prediction of interactions in a blind test

We continue the performance evaluation by testing CAPPI's ability to predict interactions in a blind test. To this end, we compute the CAPPI-Pred dataset by iteratively leaving out PPI data of one of the seven species and predicting its interactions based only on the data from the other six species. We discuss the assessment of yeast and human predicted interactomes based on the two scoring frameworks.

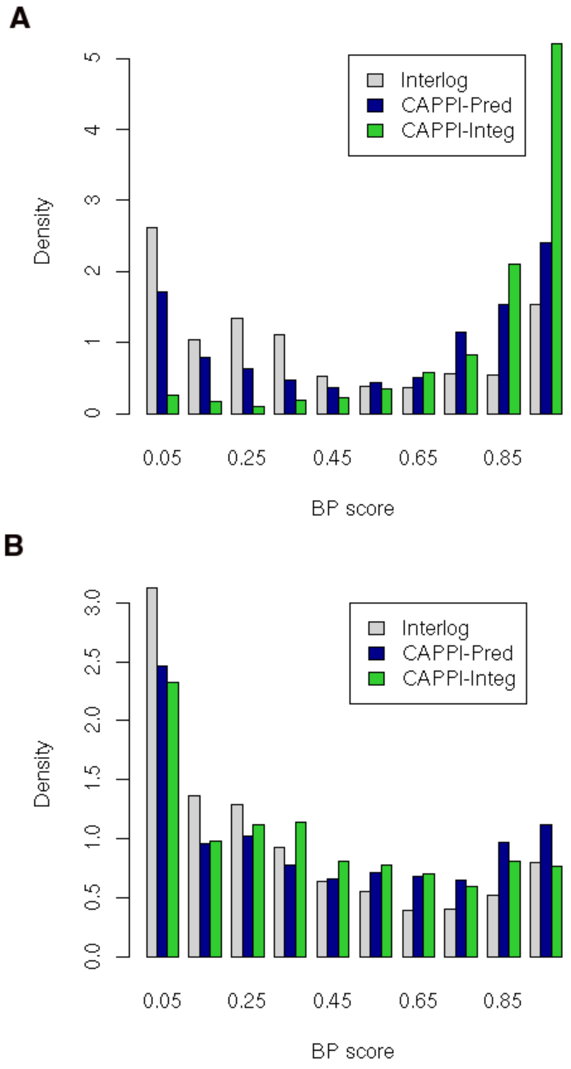


Figure 6.6: *Histogram of BP scores for the predicted yeast (A) and human (B) PPI datasets of the same size (1576 yeast PPIs and 17105 human PPIs) from the Interlog method, CAPPI-Pred and CAPPI-Integ.*

6.4.1 GO-based assessment of yeast and human predictions

Figure 6.6 shows multiple histograms summarizing the *BP* score distribution among yeast and human predictions, respectively. The sizes of the predicted dataset (1576 for yeast and 17105 for human) have been selected to allow comparison with the interlog mapping predictions (see next section for details). Interestingly, we observe that while the performance of CAPPI-Pred is lower than CAPPI-Integ in case of yeast predictions, the opposite is true for the predicted human interactome. This suggests that while the yeast input interactions are necessary for good prediction results, human input datasets, on average, bring a less notable contribution.

6.4.2 Validation based on reference datasets

In Fig. 6.7 (A) we plot the ratio of true positives and false positives as a function of the number of yeast PPIs returned by CAPPI-Pred. We evaluate the predictions separately using co-complex datasets (CAPPI-Pred Complex), gold standard binary PPI datasets (CAPPI-Pred PPI), as well as all available reference data (CAPPI-Pred All) – see Appendix A for details. An analogous study is performed for the predicted human interactome using the HPRD (complex and binary PPI) catalogues as reference (see Fig. 6.7 (B)). Note that similarly as for yeast, also for human the positive reference set is significantly smaller than the negative reference set. The joint human reference set (All) contains 57,093 protein pairs, which is less than 0.2% of the number of differentially localized pairs – consistent with the expected ratio of true interactions to all protein pairs in human, as estimated by Stumpf *et al.* (2008). The results show that CAPPI is able to infer high-scoring PPIs also in the case when no interactions from the predicted interactome are included in the training set. Most of the top predictions are confirmed by experimental data. We observe that while more yeast predictions are confirmed by co-complex pairs than by binary PPI data, the opposite is true in case of the human predictions. This can be explained by the differences in size of the respective reference datasets for the two species (see Appendix A). When all available reference data is considered (CAPPI-Pred-All), the TP/FP

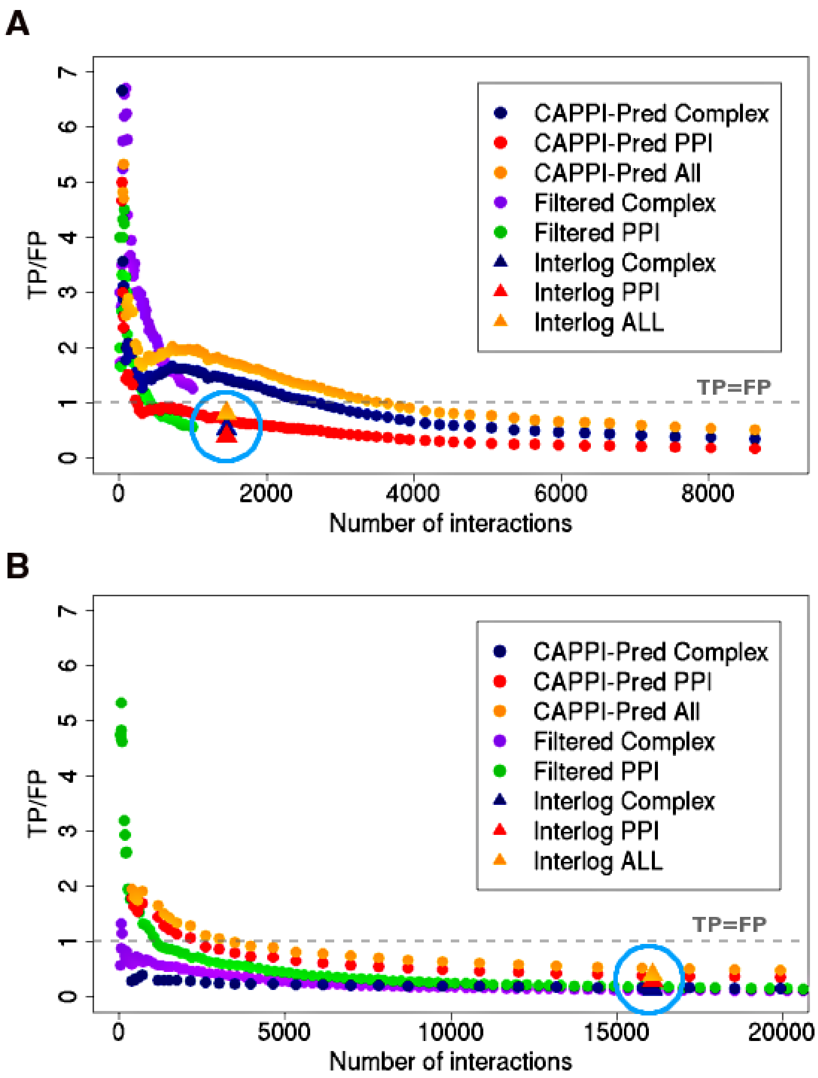


Figure 6.7: *The ratio of true positives and false positives as a function of the number of interactions in the CAPPI-Pred dataset for yeast (A) and human (B). An interaction is deemed true positive, if it is found in the reference dataset of either co-complex interactions (Complex) or binary PPIs (PPI), or in any available reference set (All). Otherwise it is considered false positive if the proteins are assigned to different cellular localization (see text and Appendix A). Plots labeled as "Filtered Complex" and "Filtered PPI" show the results of selected CAPPI predictions which are part of dense clusters – tested against either the co-complex reference (Complex) or binary reference (PPI). The gray dashed line marks the level at which the number of true positive predictions is equal to the number of false positive predictions.*

ratios for the top 5000 interactions in yeast and human are comparable (~ 0.8).

Filtering co-complex predictions Evolutionary pressures are more likely to constrain essential functional complexes than individual interactions (Beltrao & Serrano, 2007). Thus co-complex PPIs should be easier to map accurately across species. This premise was previously explored by Sharan *et al.* (2005b), who showed that screening PPI predictions against conserved clusters improves prediction specificity. In an attempt to increase the percentage of co-complex PPIs in our predictions, we filtered the CAPPI-Pred output dataset, leaving only the predicted PPIs placed within conserved dense network regions. To this end, an ancestral interaction network was computed as in Chapter 5, and clustered using the MCL algorithm to identify dense clusters. Each cluster was projected onto the network of the extant species (yeast or human) and CAPPI-Pred predictions within the projected regions were identified as a result. As shown in Fig. 6.7, this procedure significantly boosts the TP/FP ratio for both yeast and human data (see "Filtered Complex" plots in Fig. 6.7). Interestingly, while the fraction of co-complex PPIs was increased, the fraction of confirmed binary PPIs was in general lowered by the filtering (except for the top ranked human predictions), suggesting that many binary PPIs placed outside or between protein complexes are filtered out in this case. This is in line with the observations made by Yu *et al.* (2008) that binary and co-complex datasets are of complementary nature and often have small overlap.

6.5 Comparison with previous high-throughput multi-species approaches

As we acknowledged in the beginning, in the last few years different groups developed a number of interaction inference methods, very diverse in their inherent assumptions and working principles. These approaches often rely on different input data, limiting the possibility for a direct comparison. Although it is technically possible to integrate heterogeneous information in our framework, it is beyond the scope of this study. Here we select two methods which, like CAPPI, rely purely on PPI datasets and integrate and transfer interaction evidence across species. We compare our results with the interactions inferred by a domain-based maximum likelihood procedure (Liu *et al.*, 2005) and an interlog-based PPI mapping framework (Michaut *et al.*, 2008). Both of these general approaches have been implemented in numerous studies and applied to many different datasets.

Comparison with the domain-based maximum likelihood approach Liu *et al.* (2005) generalized the domain-domain interaction prediction method to multiple species and applied it to infer interactions in yeast, worm and fly (we refer to this method as the Domain-ML approach). As a final output, this approach predicts protein-protein interactions based on inferred interactions between conserved domains. Liu *et al.* trained their method using Ito, Uetz, Giot and Li experimental datasets, so the their results can be directly compared to CAPPI-Integ-3sp. Note that only the yeast interaction predictions were provided by the authors. The mean *GO* scores for Domain-ML and CAPPI are shown in Fig. 6.4. CAPPI-Integ-3sp significantly outperforms Domain-ML in terms of all three *GO* scores. The performance evaluation using gold standard data (Fig. 6.5) also indicates a higher accuracy of CAPPI compared to the domain-based approach.

Comparison with the interlog-based approach Next, we compare our results with a popular method of interlog mapping. This approach, similarly to CAPPI, relies on protein sequence similarity to transfer the interaction evidence across species. We choose for comparison the interlog mapping implementation of Michaut *et al.* (2008) and use the same input data in predicting our CAPPI-Pred dataset (for details see Appendix A). Figure 6.6 provides the distributions of *GO* scores for the Interlog and CAPPI datasets of the same size: 1576 (yeast) and 17105 (human), respectively. CAPPI predictions contain a larger fraction of high-scoring interactions (those with *GO* score > 0.8) and obtain a higher average score. The mean score for the CAPPI-predicted yeast dataset is significantly higher than that of the Interlog method (0.57 vs. 0.39). CAPPI's advantage is also apparent in case of the human predictions (mean score 0.42 vs. 0.33). Figure 6.7 shows the mean scores for the Interlog output (in blue circles), which can be compared with the CAPPI rankings. In all cases CAPPI achieves a higher fraction

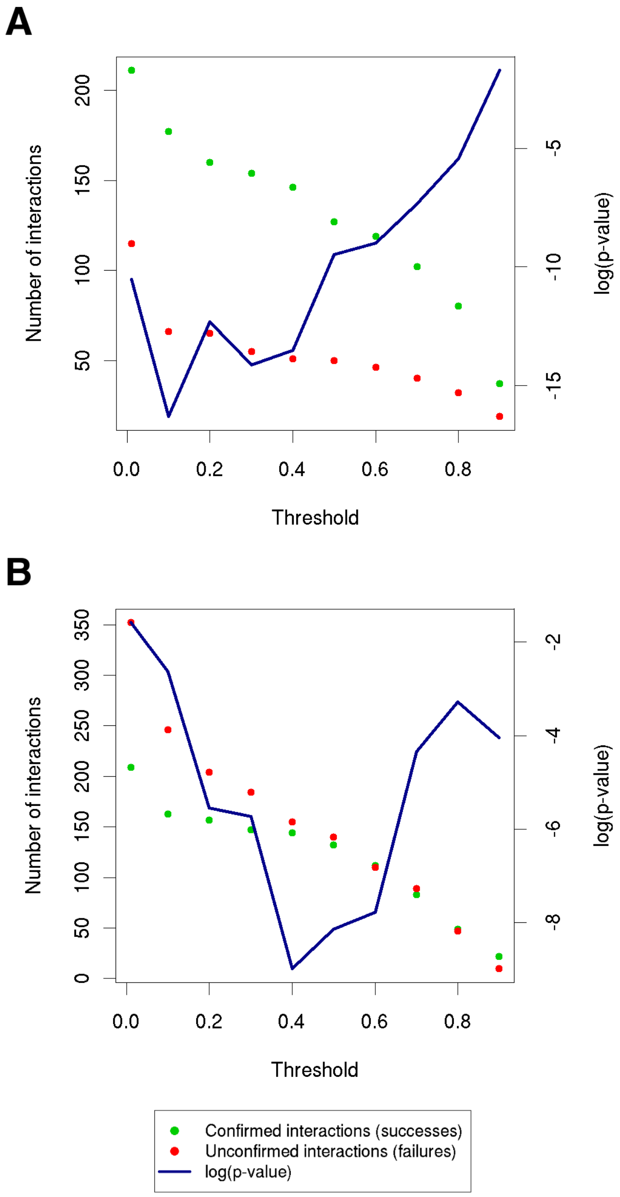


Figure 6.8: *The number of confirmed predictions, unconfirmed predictions and corresponding p-values (in logarithmic scale), as a function of the threshold, for interactions among the 26S proteasome proteins from yeast (A) and human (B). The p-values are computed based on the hypergeometric distribution where confirmed interactions are considered as successes and unconfirmed interactions are considered as failures. For both species CAPPI predictions are significant over a wide range of thresholds. The apparent threshold with the minimum p-value may serve as a point of reference at which predictions can be analyzed.*

of true positive interactions: 0.88 vs. 0.47 for the yeast co-complex predictions, 0.72 vs. 0.40 for the yeast binary PPI prediction, 0.16 vs. 0.14 for the human co-complex predictions, and 0.38 vs. 0.28 for the human binary PPI predictions. These results provide evidence that phylogeny-based mapping of PPI data from multiple homologs performs favorably to the classical interlog mapping approach. CAPPI's additional advantage lies in the provided ranking (induced by the posterior probabilities), which enables the user to easily identify the most reliable interactions. As an example, for the purpose of selecting human PPI targets for verification, one could make a heuristic decision to consider only around 3500 top predictions for which the TP/FP ratio is greater than 1 (see Fig. 6.7 (B)).

6.6 Case studies: mapping interactions within conserved functional modules

We now zoom-in on specific examples of functional units in the interactomes of human, yeast and thale cress, and analyze co-complex interactions inferred by CAPPI-Pred. In all described cases we demonstrate that the general topological features and organization of these complexes, as well as many known pairwise PPIs, can be recovered by our method based solely on data from the other species. We verify the inferred interactions against previously reported experimental data and compute a hypergeometric p-value to assess the significance of our predictions. For an example of how the threshold selection impacts the number of interactions and the resulting p-value see Fig. 6.8. Note that in the following discussion gene names are used to denote corresponding proteins.

6.6.1 Human and yeast proteasome subnetworks

The ubiquitin-proteasome pathway is essential for eliminating damaged proteins and for regulation of intra-cellular level of proteins involved in wide spectrum of cellular functions (Glickman & Ciechanover, 2002). It is conserved in eukaryotes, from yeast to human. The 26S proteasome complex contains a 20S catalytic core particle (CP), which is capped on each side by a 19S regulatory particle (RP). The structure of the 20S proteasome from yeast has been resolved (Groll

& Huber, 2005). It consists of 28 protein subunits: two α-rings ($\alpha1, \ldots, \alpha7$) and two β-rings ($\beta1, \ldots, \beta7$). The 19S proteasome can be further decomposed into two subcomplexes: the base (Rpt1-Rpt6, Rpn1, Rpn2, Rpn10 and Rpn13 – the last one probably not present in human) that binds directly to the 20S proteasome, and the lid (Rpn3, Rpn5-Rpn9, Rpn11, Rpn12 and Sem1), which is a peripheral subcomplex. In addition there is a number of transiently associated factors like p27 and S5b (the latter is apparently not present in yeast). We discuss our predictions of the 26S proteasome interactions from yeast and from human separately.

Yeast proteasome Predicted interactions in the yeast 26S proteasome are depicted in Fig. 6.9. The presented graph is split into four parts that correspond to the four subcomplexes of the proteasome: α-ring, β-ring, lid and base. The α-ring and the β-ring have a dense set of interactions. Both of them together form a clique (i.e. every two proteins are predicted to interact), with most of the interactions being supported by experimental data. The lid and base are also very well represented and connected by 16 interactions, all of which are confirmed by previous experiments. We observe also the central role of Rpn7, which is predicted to interact with every subunit in the α- and and in the β-ring, as well as with six proteins in the lid subcomplex and eight in the base. Another hub protein identified is Rpn1, which has twelve interaction partners among the alpha and beta proteins (four of which are confirmed), seven partners in the base and seven in the lid (all having experimental support). The transiently associated NAS2 (p27) is predicted to interact only with the AAA-ATPase subunits (Rpt1-Rpt6) of the base subcomplex. In general, interactions within the core subcomplexes of the yeast 26S interactome are accurately recovered based solely on data from other six species, demonstrating a high level of conservation of these PPIs. The vast majority of the 66 unconfirmed prediction are localized between the characterized subcomplexes. In fact only 7 of the 44 predicted interactions between the 20S catalytic core and and the 19S regulatory particles are backed by experimental evidence in yeast. The absence of experimental data for these PPIs in *S. cerevisiae* might be explained by insufficient coverage of the yeast interactome or by possible

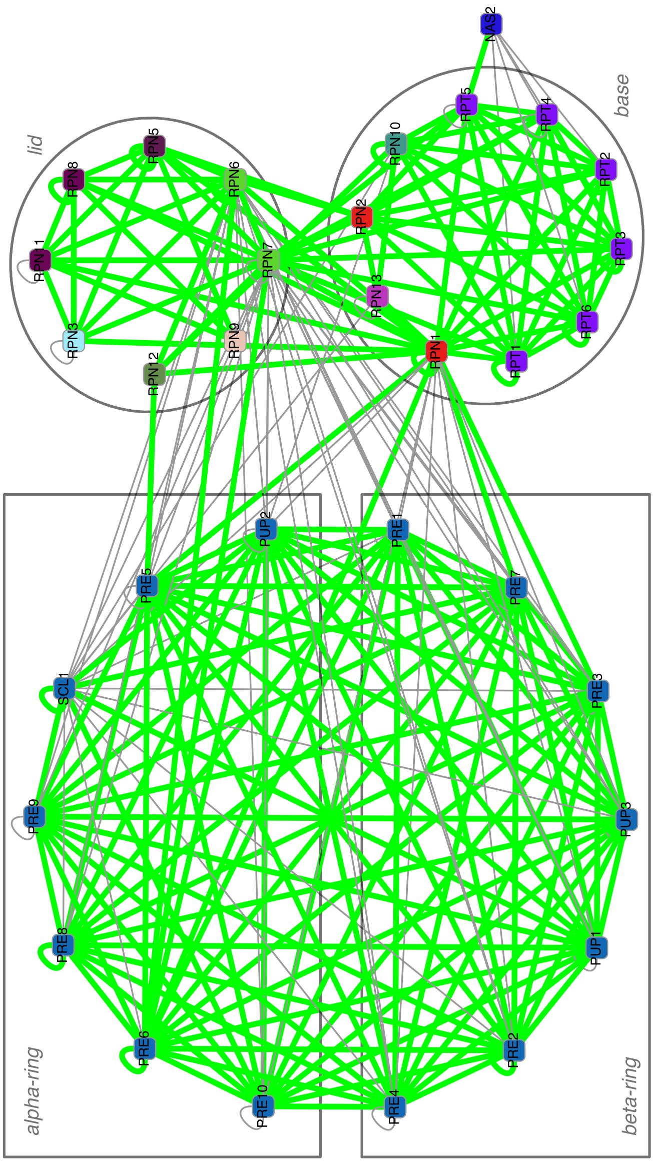


Figure 6.9: *Interaction network of the yeast 26S proteasome complex as inferred by CAPPI-Pred. Nodes represent gene products and node colors represent protein families identified by sequence clustering. 177 of the predicted interactions which have been previously detected experimentally are denoted by green edges. 66 other PPI predictions are denoted by gray edges. The p-value of the predicted network is* $4.348 * 10^{-16}$.

rewiring events which changed the topology of interactions between the conserved core subunits across species.

Human proteasome The result of our predictions of subunit-subunit interactions in the human proteasome is depicted in Fig. 6.10. Compared to the yeast proteasome map, the human subnetwork contains more previously unreported interactions, possibly due to the incompleteness of the human data. Again, the resulting graph is split into four parts, each corresponding to a distinct subcomplex. The α- and β-rings clique representation is very similar to that of the yeast proteasome. The base subcomplex also has a very dense set of interactions. Some of these PPIs, namely PSMC2–PSMC6 (Rpt1–Rpt4), PSMC2–PSMC5 (Rpt1–Rpt6) and PSMC4–PSMC4 (Rpt3–Rpt3) have only recently been reported (Chen *et al.*, 2008). Like in the case of the yeast proteasome, we notice a central position of PSMD9 (p27) with respect to AAA-ATPase subunits PSMC1-PSMC6 (Rpt1-Rpt6), which has also been reported by Chen *et al.* (2008). Due to these confirmed interactions we decided to merge this protein into the base subnetwork. We also observe that PSMD8 (Rpn12) of the lid is predicted to densely interact with the base proteins. PSMD1 (Rpn2) and PSMD2 (Rpn1) each have five predicted interactions with other lid members and none with the base. Therefore we decided to move them into the lid subnetwork. In fact PSMD1 is described as the largest non-ATPase subunit of the 19S regulator lid by the Entrez Gene database, somewhat differently from its Rpn2 homolog in yeast (usually attributed to the base subcomplex). PSMD2, like its yeast homolog Rpn1, has many predicted interactions with both subcomplexes (α and β) of the 20S proteasome, however in human they are not confirmed. We also notice a dense set of interactions between the six AAA-ATPase subunits (PSMC1-PSMC6) and the 20S catalytic core. Many of these interactions have not been previously reported in the literature and could be used as hypothesis in verifying experiments. Compared to our result for yeast, our representation of the lid subcomplex of the human 19S proteasome lacks PSMD13 (Rpn9), which we did not find in the initial Integr8 dataset. In human, we find an additional transiently associated protein PSMD5 (S5b), which binds to PSMC2.

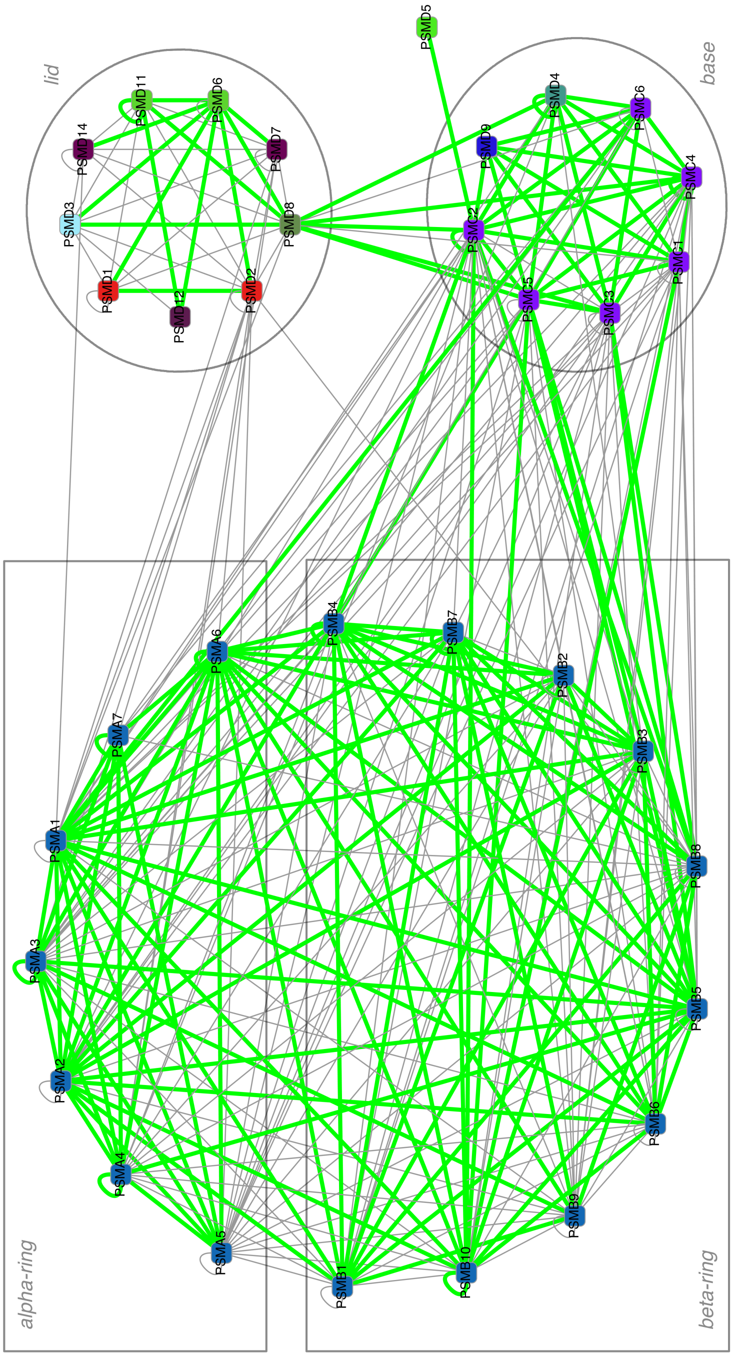


Figure 6.10: *Interaction network of the human 26S proteasome complex as inferred by CAPPI-Pred. Nodes represent gene products and node colors represent protein families identified by sequence clustering. 144 of the predicted interactions which have been previously detected experimentally are denoted by green edges. 155 other PPI predictions are denoted by gray edges. The p-value of the predicted network is* $1.614 * 10^{-6}$.

6.6.2 Human and yeast endosome subnetworks

The ESCRT complexes comprise a major pathway for the lysosomal degradation of transmembrane proteins (see Hurley & Emr, 2006). We investigate the predicted interactions for the ESCRT complexes in human and yeast and compare the obtained results with the interactions reported in the literature. The list of proteins involved in these complexes was taken from Hurley & Emr (2006).

Human ESCRT Complexes Human ESCRT co-complex interactions as predicted by our method are depicted in Fig. 6.11. CAPPI-Pred was able to recover all five complexes discussed in Hurley & Emr (2006). These complexes are: ESCRT-3 (well represented as a dense connected component with most edges reported in previous experiments), ESCRT-1, ESCRT-0, the Vps4 complex, and the ESCRT-2 complex. Interestingly, our results suggest that proteins CHMP1B and CHMP5 should be assigned to the ESCRT-3 complex. This association of CHMP1B and CHMP5 (consistent with the so called 'CHMP nomenclature') has been recently proposed by Shim *et al.* (2008). Moving on to the right side of the graph, we notice that the VPS4 proteins together with protein VTA1 form a triangle comprising of three reported interactions. A similar observation can be made for the ESCRT-0 complex (HGS, STAM1 and STAM2), except that the interaction STAM–STAM2 is not supported by previous experimental data. Also, the topology of interactions presented in Fig. 6.11 suggests an important role of the TSG101 (mammalian VPS23) protein, which joins ESCRT-1 with three other complexes (ESCRT-3, ESCRT-0 and Vps4). TSG101 also takes part in five identified interactions within the ESCRT-1 complex, all of which have backing experimental evidence in human.

Yeast ESCRT Complexes These complexes, as a result of our method, are depicted in Fig. 6.12. We find that almost all predicted interactions (except for five self loops) are supported by experimental studies. Similarly as in the human network, all five complexes discussed by Hurley & Emr (2006) can be naturally retrieved from the presented graph. Like in the human ESCRT complexes, the yeast homologs of CHMP1B (DID2) and of CHMP5 (VPS60) well fit (graph-theoretically) to ESCRT-3. This again confirms the observation stated in Shim

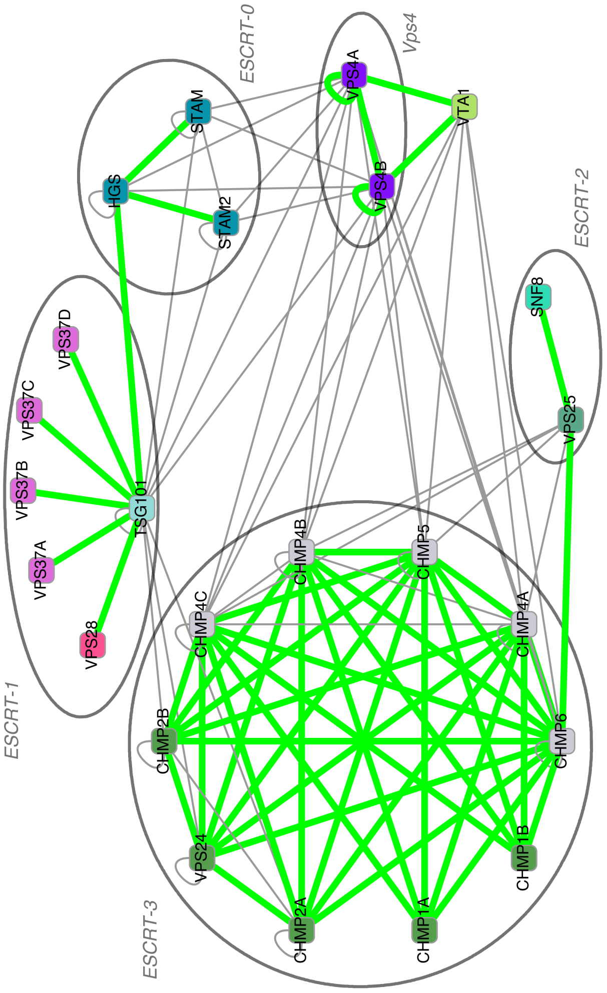


Figure 6.11: *Interaction network of the human endosome complexes as inferred by CAPPI-Pred. Nodes represent gene products and node colors represent protein families identified by sequence clustering. 49 predicted interactions which have been previously detected experimentally are denoted by green edges. 49 other PPI predictions are denoted by gray edges. The p-value of the predicted network is* $3.977 * 10^{-9}$.

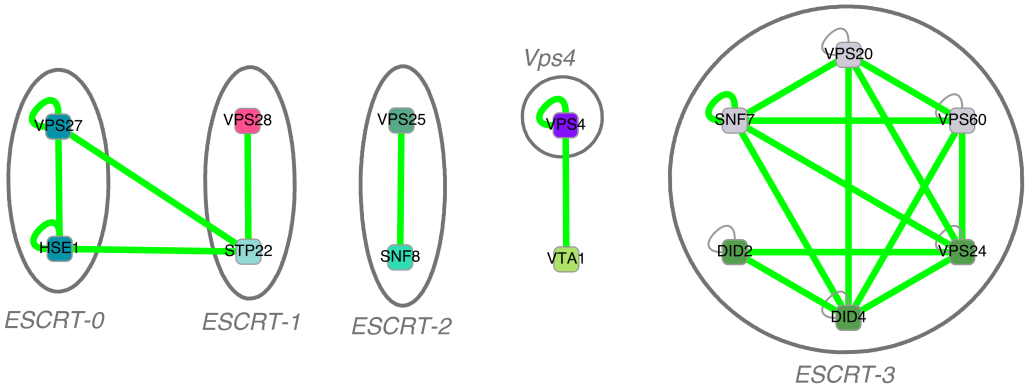


Figure 6.12: *Interaction network of the yeast endosome complexes as inferred by CAPPI-Pred. Nodes represent gene products and node colors represent protein families identified by sequence clustering. 22 of the predicted interactions which have been previously detected experimentally are denoted by green edges. 5 other PPI predictions are denoted by gray edges. The p-value of the predicted network is* $9.571 * 10^{-11}$.

et al. (2008), where the authors call these two proteins 'proposed regulatory members' of ESCRT-3. Similarly to the human endosome network, the topology of the identified interaction network suggests that VPS23 (STP22) may play an important role mediating the interactions between complexes, although at the selected threshold we did not identify its interactions with the ESCRT-3 and Vps4 complexes (as we did in the human example).

6.6.3 Human mRNA decay complexes

Next we investigated CAPPI's interaction predictions between proteins involved in human mRNA degradation (see Lehner & Sanderson, 2004). The subgraph of predicted interactions is presented in Fig. 6.13. We have a very good coverage of the human exosome complex represented by six RNase PH domain subunits (EXOSC4 (Rrp41), EXOSC5 (Rrp46), EXOSC6 (Mtr3), EXOSC7 (Rrp42), EXOSC8 (Oip2), EXOSC9 (PMScl-75)), three S1 RNA-binding domain subunits (EXOSC1 (Csl4), EXOSC2 (Rrp4), EXOSC3 (Rrp40)), and an RNase D-like subunit EXOSC10 (PMScl-100). This complex comes out as a complete subgraph (a clique)

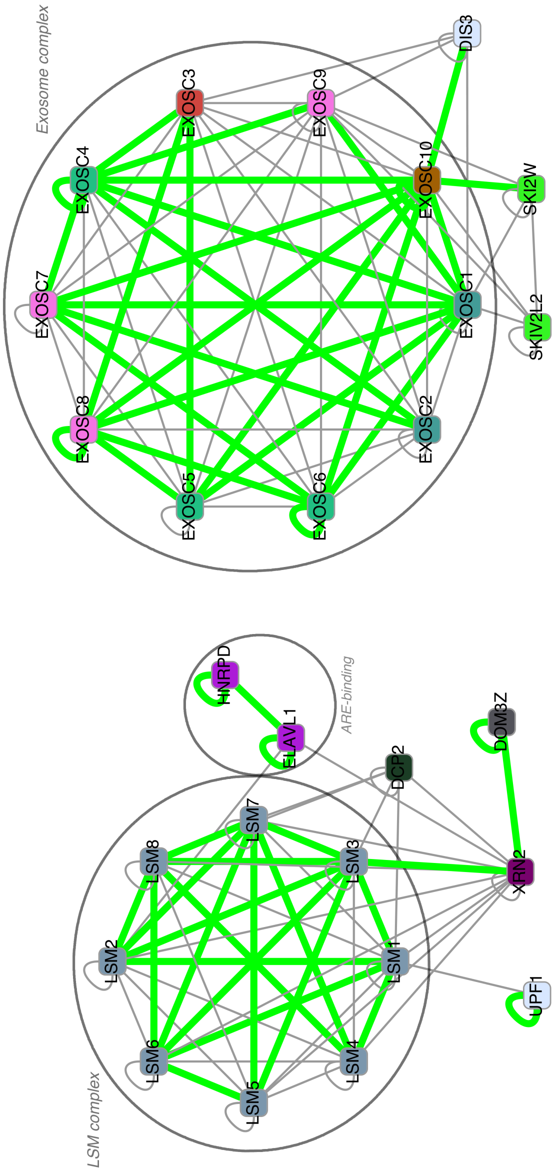


Figure 6.13: *Interaction network of the human mRNA decay complexes as inferred by CAPPI-Pred. Nodes represent gene products and node colors represent protein families identified by sequence clustering. 53 of the predicted interactions which have been previously detected experimentally are denoted by green edges. 76 other PPI predictions are denoted by gray edges. The p-value of the predicted network is* $1.868 * 10^{-15}$.

with no interactions with the other two complexes. The role of most of the subunits of the complex, in terms of interacting partners, is quite comparable. One of the exceptions is the EXOSC9 (PMScl-75) protein which is the only RNase PH domain subunit predicted to interact with DIS3 and two helicases (SKI2W and SKIV2L2). Other exosome complex members interacting with DIS3 are S1 RNA-binding subunits EXOSC1 (Csl4) and EXOSC3 (Rrp40), as well as EXOSC10. EXOSC1 and EXOSC10 also have predicted interactions with helicases SKI2W and SKIV2L2. In general, data on interactions of the peripheral subunits with the exosome complex are scarce, as reported in Lehner & Sanderson (2004), which makes our predictions a potentially valuable target for experimental verification.

The second complex which comes out as a dense subgraph in our network is the LSM complex. It consists of eight proteins (LSM1-8), forming a clique of predicted interactions, many of which are confirmed experimentally (see Lehner & Sanderson, 2004, Fig. 3A). The two proteins with the largest number of confirmed interactions within the complex are LSM3 and LSM7. Both of these proteins have confirmed PPIs with six out of seven other LSM members (additional PPIs predicted by our method are LSM3–LSM4 and LSM7–LSM1).

The third complex which can be retrieved from the network in Fig. 6.13 consists of two AU-rich element ARE-binding proteins (ELAVL1 (Hur) and HNRPD (Auf1)). All three interactions predicted inside this complex are confirmed by recent experimental data (see David *et al.*, 2007). Among the unverified predictions is an interaction of this complex with the LSM complex (via LSM2) and with another mRNA decay factor XRN2.

6.6.4 *A. thaliana* SWI/SNF chromatin remodeling complex

In yeast and mammals, ATP-dependent chromatin remodeling complexes belonging to the SWI/SNF family play an essential role in the regulation of transcription. In Arabidopsis chromatin remodeling complexes are known to a much smaller extent. No plant SWI/SNF complex has been established and characterized to date, but it is highly probable that such complexes exist in plants (see Jerzmanowski, 2007). For this reason it seems desirable to employ a computational approach

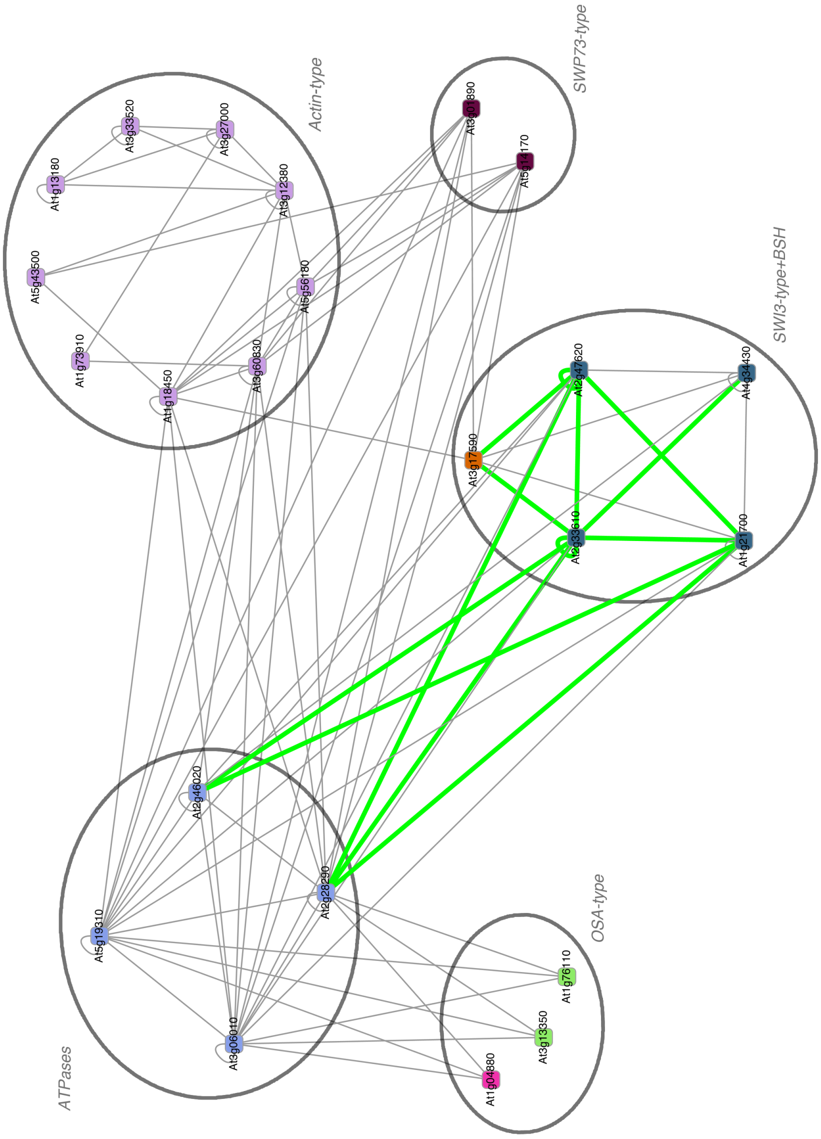


Figure 6.14: *Interaction network of the putative SWI/SNF complex in Arabidopsis as inferred by CAPPI-Pred. Nodes represent gene products and node colors represent protein families identified by sequence clustering. 13 of the predicted interactions which have been previously detected experimentally are denoted by green edges. 83 other PPI predictions are denoted by gray edges. The p-value of the predicted network is* $6.381 * 10^{-10}$.

for predicting interactions in the plant SWI/SNF putative complex and generate plausible working hypothesis. We present a zoom-in view of the SWI/SNF putative complex in Fig. 6.14. A larger zoom-out view containing other homologs of the putative SWI/SNF complex members is presented in Fig. 6.15.

The graph in Fig. 6.14 contains the core SWI/SNF proteins – the SWI3-type proteins: At2g47620 (SWI3A), At2g33610 (SWI3B), At1g21700
(SWI3C), At4g34430 (SWI3D), together with the SNF5-type protein At3g17590 (BSH). This core is presented at the bottom of the graph. In addition to the above proteins we considered four groups of Arabidopsis proteins which are reported to play a putative role in chromatin remodeling in this plant (see Jerzmanowski, 2007). These are: four ATPases which are reported by Jerzmanowski (2007) as potential members of the SWI/SNF complex (At2g46020 (BRM), At2g28290 (SYD), At3g06010 (Chr 12), At5g19310 (Chr 23)); two SWP73-type proteins (At3g01890 (SWP73A), and At5g14170 (SWP73B)); nine actin-related proteins (At3g27000 (ARP2), At1g13180 (ARP3), At1g18450 (ARP4), At1g73910 (ARP4A), At3g12380 (ARP5), At3g33520 (ARP6), At3g60830 (ARP7), At5g56180 (ARP8) and At5g43500 (ARP9)); and three OSA-type proteins (At1g04880, At1g76110, and At3g13350). We excluded from the graph proteins which did not show any predicted interactions. Altogether we identified 13 of 14 known interactions between the proteins visualized in Fig. 6.14 – the missing one is At3g01890–At1g21700 (see Jerzmanowski, 2007). We notice some interesting peculiarities of the presented network. All SWI3-type proteins, with the exception of At4g34430 (SWI3D), are predicted to interact with the four ATPases. Only one actin-type protein (At1g18450) has a predicted interaction with the SWI/SNF core and only two more (At3g60830 and At5g56180) can be associated with the complex through member ATPases. The ability to make distinctions within family members is an important indicator of the performance of our approach. A method like CAPPI, which bases its prediction on evidence from all homologs, is likely to always assign the same interactions to all family members in one species. The above examples demonstrate that this is not the case, and that the phylogenetic information used by our method allows it to distinguish between family members when distributing the interaction evidence.

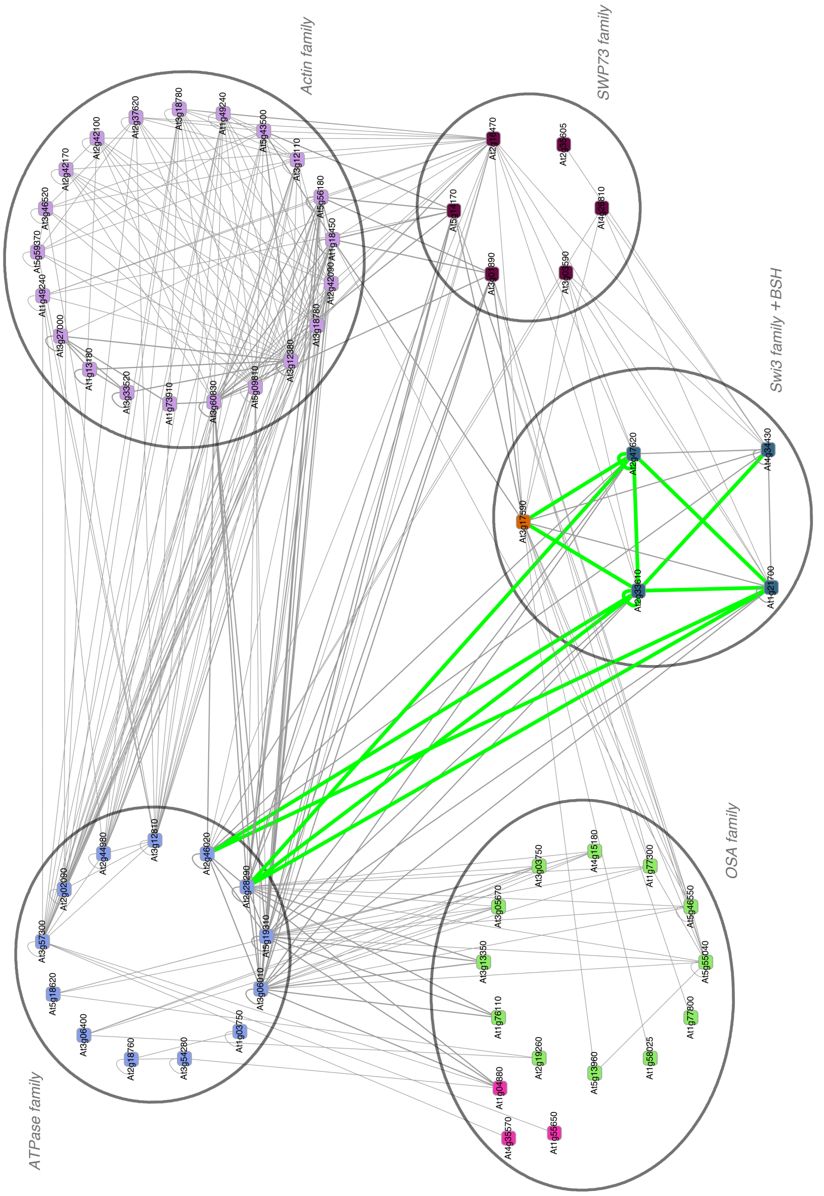


Figure 6.15: *An extended view of the interaction network of the putative SWI/SNF complex in Arabidopsis as inferred by CAPPI-Pred. Nodes represent gene products and node colors represent protein families identified by sequence clustering. 13 of the predicted interactions which have been previously detected experimentally are denoted by green edges. 319 other PPI predictions are denoted by gray edges. The p-value of the predicted network is* $3.411 * 10^{-9}$.

These observations are strengthened when we consider the larger family-oriented view of the SWI/SNF-related network in Fig. 6.15. This graph was obtained from the one in Fig. 6.14 by expanding the set of proteins to all members of the considered protein families (once again, proteins without any interactions were removed). Interestingly, the four peripheral families represented in the graph can be divided into smaller subfamilies based on the interactions partners of their members. Specifically, of the 14 ATPases presented in the larger graph only the four above described are predicted to interact directly with the core of the SWI/SNF complex. Two of them (At2g46020 (BRM) and At2g28290 (SYD)) have confirmed interactions while for the other two (At3g06010 (Chr 12), At5g19310 (Chr 23)) interaction hypothesis based on sequence similarity were formulated (Jerzmanowski, 2007). In fact the entire ATPase family, as detected by our method, contains 48 Arabidopsis proteins (a vast majority not having any predicted interactions to other proteins in the SWI/SNF subnetwork), which makes the presented predictions even more significant. These specific cases of confirmed predictions let us suggest that some of the distinctive members of the other protein families predicted to interact with the putative SWI/SNF complex (At1g18450 and six OSA family members interacting with At3g17590, five SWP73 family members interacting either with At3g17590 or at least one of the SWI3-type proteins, as well as five other actin family members interacting with ATPases At2g46020 and At2g28290), may pose valuable targets for future experimental validation.

Figure 6.15 may also serve as a handy device which qualitatively visualizes interaction profiles between sub-families of proteins. We notice many interactions between part of the actin family and some of the ATPases, as well as three SWP73-type proteins. Also part of the OSA family interacts with the ATPases and with the SWP73-type group. Finally, the SWI3-type proteins also interact with selected ATPases and proteins from the SWP73 family. Interestingly SWP73-type proteins interact with the same ATPases as the SWI3-type proteins. Interactions between other family pairs are not predicted. Another interesting feature of the presented network is a clique-like structure of the actin family of proteins, which may indicate a separate protein complex formed by its members.

Chapter 7

Parameter estimation *via* expectation maximization

In the previous chapters we assumed that the parameters $\Theta = (p_s, \delta_s, p_d, \delta_d, p_1)$ of our model are known. In practice they were derived based on previous studies and adjusted empirically. In this chapter we present a procedure for learning (or estimating) the parameters from data. The method follows the expectation maximization (EM) scheme (Dempster *et al.*, 1977). It is designed towards determining parameter values that maximize the probability of the observed data when values of some of the random variables are unknown (hidden). We first describe the basic principles of the maximum likelihood (ML) paradigm and discuss the EM scheme for an abstract statistical model. Next we derive the steps of the EM procedure specifically for our model and show how they can be efficiently computed.

7.1 Maximum likelihood parameter estimation

Suppose we observe a set of n data points $\mathbf{x} = \{x_1, \ldots, x_n\}$ drawn from a given statistical model. Suppose also that the model is determined by a set of parameters Θ. Assuming that the data points are drawn independently, we can write the probability of the dataset given Θ as:

$$P(\mathbf{x}|\Theta) = \prod_{i=1}^{n} P(x_i|\Theta)$$

The above probability is also referred to as the likelihood $L(\Theta|\mathbf{x}) \equiv P(\mathbf{x}|\Theta)$ of the parameters Θ given the data $\mathbf{x}$. It is often the case that Θ is unknown and we want to deduce it from a given dataset. A natural criterion one may assume when deciding on the best estimate of the parameters is to choose Θ^* which achieves the maximum likelihood:

$$\Theta^* = \operatorname*{argmax}_{\Theta} L(\Theta|\mathbf{x}) = \operatorname*{argmax}_{\Theta} \prod_{i=1}^{n} P(x_i|\Theta).$$

In practice it is often more convenient to maximize the log-likelihood function:

$$\Theta^* = \operatorname*{argmax}_{\Theta} \log L(\Theta|\mathbf{x}) = \operatorname*{argmax}_{\Theta} \sum_{i=1}^{n} \log P(x_i|\Theta).$$

In certain cases obtaining the maximum likelihood estimate might be straightforward by analytical means (for examples see Bertsekas & Tsitsiklis, 2008, p. 462), whereas for some models it might turnout difficult. One often encountered complication considered here is the case when the observed data contain hidden values. The EM procedure described below is a popular approach for estimating parameters in such situations.

7.2 EM basics

The EM approach is used to estimate the maximum likelihood parameters of some statistical model in case of incomplete data. Let's assume that besides the observed data $\mathbf{x} \in \mathcal{X}$ we also have some hidden data $\mathbf{y} \in \mathcal{Y}$. We assume a joint density function of the complete data (observed and hidden):

$$P(\mathbf{x}, \mathbf{y}|\Theta) = P(\mathbf{y}|\mathbf{x}, \Theta)P(\mathbf{x}|\Theta). \tag{7.1}$$

To find the log-likelihood of the observed data $\mathbf{x}$, we integrate over all possible instances of $\mathbf{y} \in \mathcal{Y}$:

$$\log P(\mathbf{x}|\Theta) = \int_{\mathcal{Y}} \log P(\mathbf{x}, \mathbf{y}|\Theta) d\mathbf{y}. \tag{7.2}$$

Our aim is to find Θ that maximizes (7.2), i.e. a maximum likelihood estimate for Θ. Suppose now that we are given a (presumably suboptimal) set of parameters Θ^0. We derive an iterative procedure which aims at providing a better set of parameters Θ^{t+1}, given the parameter set Θ^t.

Using (7.1), we can write the log-likelihood function as

$$\log P(\mathbf{x}|\Theta) = \log P(\mathbf{x}, \mathbf{y}|\Theta) - \log P(\mathbf{y}|\mathbf{x}, \Theta)$$

for any given instance of $\mathbf{y}$. By integrating over $\mathbf{y} \in \mathcal{Y}$ and weighting by the probability $P(\mathbf{y}|\mathbf{x}, \Theta^t)$ we obtain

$$\log P(\mathbf{x}|\Theta) = \int_{\mathcal{Y}} P(\mathbf{y}|\mathbf{x}, \Theta^t) \log P(\mathbf{x}, \mathbf{y}|\Theta) d\mathbf{y} - \int_{\mathcal{Y}} P(\mathbf{y}|\mathbf{x}, \Theta^t) \log P(\mathbf{y}|\mathbf{x}, \Theta) d\mathbf{y}.$$

Notice that the first term on the right side of the equation is the expected value of the log-likelihood of the complete data (both observed and hidden), with respect to the unknown data $\mathbf{y}$, conditional on the observed data $\mathbf{x}$, and the current parameter estimates Θ^t. We will refer to this expectation as $\mathcal{Q}(\Theta|\Theta^t)$:

$$\mathcal{Q}(\Theta|\Theta^t) = \mathbb{E}[\log P(\mathbf{x}, \mathbf{y}|\Theta)|\mathbf{x}, \Theta^t] = \int_{\mathcal{Y}} P(\mathbf{y}|\mathbf{x}, \Theta^t) \log P(\mathbf{x}, \mathbf{y}|\Theta) d\mathbf{y}. \tag{7.3}$$

We want to maximize $\log P(\mathbf{x}|\Theta)$, so we should find Θ with a greater likelihood than Θ^t. We can write the change in likelihood as

$$\begin{aligned}
&\log P(\mathbf{x}|\Theta) - \log P(\mathbf{x}|\Theta^t) = \\
&= \int_{\mathcal{Y}} P(\mathbf{y}|\mathbf{x}, \Theta^t) \log P(\mathbf{x}, \mathbf{y}|\Theta) d\mathbf{y} - \int_{\mathcal{Y}} P(\mathbf{y}|\mathbf{x}, \Theta^t) \log P(\mathbf{x}, \mathbf{y}|\Theta^t) d\mathbf{y} \\
&- \int_{\mathcal{Y}} P(\mathbf{y}|\mathbf{x}, \Theta^t) \log P(\mathbf{y}|\mathbf{x}, \Theta) d\mathbf{y} + \int_{\mathcal{Y}} P(\mathbf{y}|\mathbf{x}, \Theta^t) \log P(\mathbf{y}|\mathbf{x}, \Theta^t) d\mathbf{y} \\
&= \mathcal{Q}(\Theta|\Theta^t) - \mathcal{Q}(\Theta^t|\Theta^t) + \int_{\mathcal{Y}} P(\mathbf{y}|\mathbf{x}, \Theta^t) \log \frac{P(\mathbf{y}|\mathbf{x}, \Theta^t)}{P(\mathbf{y}|\mathbf{x}, \Theta)} d\mathbf{y}.
\end{aligned} \tag{7.4}$$

Notice that the last term is the Kullback-Leibler (K-L) divergence, also known as the relative entropy of $P(\mathbf{y}|\mathbf{x}, \Theta^t)$ and $P(\mathbf{y}|\mathbf{x}, \Theta)$. It is easy to show that it is always non-negative (follows directly from Jensen's inequality). For a real-valued ϕ that is twice differentiable and convex, and a random variable Z, we have (see Bertsekas & Tsitsiklis, 2008)

$$\mathbb{E}[\phi(Z)] \geq \phi(\mathbb{E}Z).$$

To show that the K-L divergence is non-negative, let us take two probability distributions $p(\mathbf{y})$ and $q(\mathbf{y})$ for $\mathbf{y}$. Let the random variable $Z = Z(\mathbf{y}) = q(\mathbf{y})/p(\mathbf{y})$ and $\phi(Z) = -\log(Z)$. We obtain

$$\begin{aligned} \int_{\mathcal{Y}} p(\mathbf{y}) \log \frac{p(\mathbf{y})}{q(\mathbf{y})} d\mathbf{y} &\geq -\log \int_{\mathcal{Y}} p(\mathbf{y}) \frac{q(\mathbf{y})}{p(\mathbf{y})} d\mathbf{y} \\ \Longrightarrow \int_{\mathcal{Y}} p(\mathbf{y}) \log \frac{p(\mathbf{y})}{q(\mathbf{y})} d\mathbf{y} &\geq 0. \end{aligned}$$

Thus, for (7.4) to be positive, it is sufficient to choose Θ^{t+1} for which $\mathcal{Q}(\Theta^{t+1}|\Theta^t) > \mathcal{Q}(\Theta^t|\Theta^t)$. Here we consider the classical formulation of EM and find Θ^{t+1} that maximizes $\mathcal{Q}$:

$$\Theta^{t+1} = \underset{\Theta}{\operatorname{argmax}}\, \mathcal{Q}(\Theta|\Theta^t).$$

In general, the EM procedure iterates the following two steps:

E-step: Calculate the $\mathcal{Q}$ function (7.3).

M-step: Find Θ^{t+1} that maximizes $\mathcal{Q}(\Theta|\Theta^t)$ with respect to Θ.

The likelihood will increase in each iteration until it reaches some local or possibly global maximum. In practice, to increase the chances of finding the globally optimal solution, one often repeats the procedure starting from different values of Θ^0.

The E-step and the M-step have to be worked out specifically for a given problem. In some cases they have an analytically tractable form and can be efficiently computed. The classic literature examples of specific EM procedures include an algorithm for estimating the parameters of a mixture of densities (Dempster *et al.*, 1977) and the Baum-Welch algorithm (Baum *et al.*, 1970) for learning the parameters of a Hidden Markov Model (HMM).

In the next section we show how the EM scheme can be performed for our model. We derive the E and M steps which are similar to those of the Baum-Welch algorithm as formulated in Durbin *et al.* (1998). Pearl's message passing algorithm is used in place of the forward-backward procedure for HMMs. We have later learned of the work of Lauritzen (1995), which applies the Lauritzen-Spiegelhalter message passing algorithm to estimate parameters in general Bayesian network models (we thank Dr. Piotr Pokarowski for bringing this reference to our attention). However, in the context of the present study, the simple application of the original MP algorithm developed by Pearl is sufficient.

7.3 EM for the CAPPI model

Recall that in our model we have three conceptual types of random variables: those corresponding to experimental outcomes (detected PPIs and pairs not observed to interact), those corresponding to extant interactions, and those corresponding to predecessor interactions. Only the experimental outcomes are assumed to be known and instantiated to either 1 (when the interaction was observed in a given experiment) or 0 (a potential interaction was not observed in a given experiment). All extant and ancestral interactions are unknown and thus the corresponding random variables are not instantiated (hidden).

Let us denote by $\mathcal{X}$ the set of all possible states of experimental outcomes, and by $\mathbf{x} = \{x_1, \ldots, x_n\} \in \mathcal{X}$ the particular instance observed in the data, where x_i is equal to 1 when the i-th experimental reading (a particular experimental observation for a given protein pair) was positive and 0 otherwise. Similarly denote by $\mathbf{y} = \{y_1, \ldots, y_m\} \in \mathcal{Y}$ one particular instance of all hidden interactions chosen from the set $\mathcal{Y}$ of all possible states of these random variables. In this case y_i equals 1 if the i-th pair of proteins interact with each other and 0 otherwise.

Let us now consider all direct parent-child pairs in our Bayesian network model, i.e. all pairs of random variables (p, c) where p is the unique direct predecessor of c. There are three general categories of these pairs corresponding either to speciations, duplications or to experimental outcomes. Formally each pair $(p, c) \in S \cup D \cup E$, where S, D and E are mutually exclusive. S is the set of pairs (a, b) such that b was formed by speciation from a. D is the set of pairs (a, b) such that b was formed by duplication from a. Finally E is the set of pairs (a, b) such that b denotes an experimental outcome for a. Considering these categories and possible values for the random variable pairs, we distinguish 12 types of parent-child transition events. Formally an event type e is an ordered triplet $e \in \{spec, dup, exp\} \times \{0, 1\} \times \{0, 1\}$. The first coordinate denotes the general category of the event – either speciation, duplication, or experimental outcome, respectively. The second and third coordinates denote the value of the parent (p) and the child (c), respectively. We write $e[i]$ for the i-th coordinate of e. It will be convenient to associate with each type of transition events a set of all pairs of random variables in the model which can be of this type. We denote this set by $C(e)$. Depending on the first coordinate $e[1]$ of e we have:

$$C(e) = \begin{cases} S, & \text{if } e[1] = spec \\ D, & \text{if } e[1] = dup \\ E, & \text{if } e[1] = exp \end{cases}$$

Let us also define the event $1 - e$ complementary to e: $1 - e \overset{\text{def}}{=} (e[1], e[2], 1 - e[3])$. We denote the probability of transition of type e by $p_e = P(c = e[3]|p = e[2])$, for a pair of random variables $(p, c) \in C(e)$. We can denote each of the parameters of our model using the new notation. For example, p_s – the probability of retaining an interaction during speciation – can be written as: $p_s \equiv p_{(s,1,1)} \equiv P(c = 1|p = 1)$, for $(c, p) \in S$. The transition event complementary to $(s, 1, 1)$ would be $(s, 1, 0)$ (interaction loss during speciation) with the probability $1 - p_s \equiv p_{(s,1,0)} \equiv P(c = 0|p = 1)$, for $(p, c) \in S$. As previously in this chapter we denote the vector of parameters by Θ. We write Θ^t and Θ^{t+1} for the current and the next estimates of parameters within the EM procedure, respectively. Note that the probability of observing an existent interaction in an experiment, as well as the probability of observing a false interaction, are treated on the same basis as the evolutionary parameters – they can also be estimated in the EM procedure. It is also possible to introduce separate parameters for each input dataset to denote its specific reliability.

Depending on the value of the hidden interactions, we may have a different number of transition events of each type. We denote by $A_e(\mathbf{y})$ the number of transition events of type e given the state of the hidden variables $\mathbf{y}$. Note that formally, for events corresponding to experimental outcomes, A_e also depends on the values $\mathbf{x}$ of the observed variables. We do not write it explicitly since $\mathbf{x}$ is known. Let us also denote by $A_1(\mathbf{y})$ the number of interactions in the ancestral network $G_{1,0}$ and by $A_0(\mathbf{y})$ the number of non-interacting pairs in the ancestral network.

7.3.1 The $\mathcal{Q}$ function

We now derive the $\mathcal{Q}$ function for our model:

$$\mathcal{Q}(\Theta|\Theta^t) = \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}, \Theta^t) \log P(\mathbf{x}, \mathbf{y}|\Theta). \tag{7.5}$$

The probability of the data given the parameters Θ can be expressed as the product of the probabilities of the initial interactions and the transition probabilities:

$$P(\mathbf{x}, \mathbf{y}|\Theta) = p_1^{A_1(\mathbf{y})} p_0^{A_0(\mathbf{y})} \prod_e p_e^{A_e(\mathbf{y})}$$

Taking the logarithm we obtain:

$$\log P(\mathbf{x}, \mathbf{y}|\Theta) = \sum_e A_e(\mathbf{y}) \log p_e + A_1(\mathbf{y}) \log p_1 + A_0(\mathbf{y}) \log p_0$$

which we can plug into (7.5):

$$\begin{aligned}
\mathcal{Q}(\Theta|\Theta^t) &= \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}, \Theta^t) \left[\sum_e A_e(\mathbf{y}) \log p_e + A_1(\mathbf{y}) \log p_1 + A_0(\mathbf{y}) \log p_0 \right] \\
&= \sum_{\mathbf{y}} \sum_e P(\mathbf{y}|\mathbf{x}, \Theta^t) A_e(\mathbf{y}) \log p_e + \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}, \Theta^t) A_1(\mathbf{y}) \log p_1 \\
&+ \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}, \Theta^t) A_0(\mathbf{y}) \log p_0 \\
&= \sum_e \log p_e \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}, \Theta^t) A_e(\mathbf{y}) + \log p_1 \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}, \Theta^t) A_1(\mathbf{y}) \\
&+ \log p_0 \sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x}, \Theta^t) A_0(\mathbf{y}).
\end{aligned} \tag{7.6}$$

Each term of the Ω function is expressed in terms of one of the expectations which we calculate in the E-step of the algorithm.

7.3.2 The E-step

The E-step of the algorithm consists of calculating the following expectations:

$$\sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x},\Theta^t)A_e(\mathbf{y}) = \mathbb{E}(A_e), \text{ for each transition type } e \tag{7.7}$$

$$\sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x},\Theta^t)A_1(\mathbf{y}) = \mathbb{E}(A_1) \tag{7.8}$$

$$\sum_{\mathbf{y}} P(\mathbf{y}|\mathbf{x},\Theta^t)A_0(\mathbf{y}) = \mathbb{E}(A_0) \tag{7.9}$$

Let's start with (7.7). Summing over all pairs of random variables (p, c) for which transition e can take place we have:

$$\begin{aligned}
\mathbb{E}(A_e) &= \sum_{(p',c')\in\{(p=e[2],c=e[3]):(p,c)\in C(e)\}} P(p',c'|\mathbf{x},\Theta^t) \\
&= \sum_{(p',c')} \frac{P(p',c',\mathbf{x}|\Theta^t)}{P(\mathbf{x}|\Theta^t)} \\
&= \sum_{(p',c')} \frac{P(c',\mathbf{x}|p',\Theta^t)P(p'|\Theta^t)}{P(\mathbf{x}|\Theta^t)} \\
&= \sum_{(p',c')} \frac{P(c',\mathbf{x}|p',\Theta^t)P(p'|\Theta^t)P(p',\mathbf{x}|\Theta^t)}{P(\mathbf{x}|\Theta^t)P(p',\mathbf{x}|\Theta^t)} \\
&= \sum_{(p',c')} \frac{P(c',\mathbf{x}|p',\Theta^t)P(p'|\Theta^t)P(p'|\mathbf{x},\Theta^t)P(\mathbf{x}|\Theta^t)}{P(\mathbf{x}|\Theta^t)P(\mathbf{x}|p',\Theta^t)P(p'|\Theta^t)} \\
&= \sum_{(p',c')} \frac{P(p'|\mathbf{x},\Theta^t)P(\mathbf{x},c'|p',\Theta^t)}{P(\mathbf{x}|p',\Theta^t)} = (*).
\end{aligned}$$

As before in Chapter 4, we split the evidence $\mathbf{x}$ into the evidence below p' ($\mathbf{d}_{p'}$) and the evidence above p' ($\mathbf{n}_{p'}$) and take advantage of the implied conditional independencies:

$$\begin{aligned}(*) &= \sum_{(p',c')} \frac{P(p'|\mathbf{x},\Theta^t)P(\mathbf{d}_{p'},\mathbf{n}_{p'},c'|p',\Theta^t)}{P(\mathbf{d}_{p'},\mathbf{n}_{p'}|p',\Theta^t)} \\ &= \sum_{(p',c')} \frac{P(p'|\mathbf{x},\Theta^t)P(\mathbf{d}_{p'},c'|p',\Theta^t)P(\mathbf{n}_{p'}|p',\Theta^t)}{P(\mathbf{d}_{p'}|p',\Theta^t)P(\mathbf{n}_{p'}|p',\Theta^t)} \\ &= \sum_{(p',c')} \frac{P(p'|\mathbf{x},\Theta^t)P(\mathbf{d}_{p'},c'|p',\Theta^t)}{P(\mathbf{d}_{p'}|p',\Theta^t)}, \end{aligned} \tag{7.10}$$

where $P(\mathbf{d}_{p'},c'|p',\Theta^t)$ can be written as:

$$\begin{aligned}P(\mathbf{d}_{p'},c'|p',\Theta^t) =& P(\mathbf{d}_{c'},\mathbf{d}_{p'-c'},c'|p',\Theta^t) = P(\mathbf{d}_{p'-c'}|p',\Theta^t)P(\mathbf{d}_{c'},c'|p',\Theta^t) \\ =& P(\mathbf{d}_{p'-c'}|p',\Theta^t)P(\mathbf{d}_{c'}|c',\Theta^t)P(c'|p',\Theta^t),\end{aligned}$$

where $\mathbf{d}_{c'}$ is the observed evidence below c' and $\mathbf{d}_{p'-c'}$ is the evidence below p' which is not below c'. We observe that each of the probabilities in (7.10) can be easily computed as part of the Pearl's message passing algorithm, either as the posterior probabilities or from the appropriate λ messages and λ values (see Chapter 4). Expression (7.8) and the complementary expression (7.9) can also be written as sums of expectations which are easily calculated using the MP algorithm.

7.3.3 The M-step

The M-step of the algorithm determines new parameter values that maximize the $\mathcal{Q}$ function. The terms of the expression (7.6) can be maximized separately with respect to one of the model parameters:

$$\begin{cases}\mathbb{E}(A_e(\mathbf{y}))\log(p_e) + \mathbb{E}(A_{1-e}(\mathbf{y}))\log(1-p_e), \text{ for each pair } (e, 1-e) \\ \mathbb{E}(A_1(\mathbf{y}))\log(p_1) + \mathbb{E}(A_0(\mathbf{y}))\log(p_0)\end{cases}$$

For a given e, we find p_e^* for which the derivative is equal to 0:

$$\begin{aligned}\frac{\partial\mathcal{Q}}{\partial p_e} &= \frac{\mathbb{E}(A_e)}{p_e^*} - \frac{\mathbb{E}(A_{1-e})}{1-p_e^*} = 0 \\ p_e^* &= \frac{\mathbb{E}(A_e)}{\mathbb{E}(A_e)+\mathbb{E}(A_{1-e})}.\end{aligned}$$

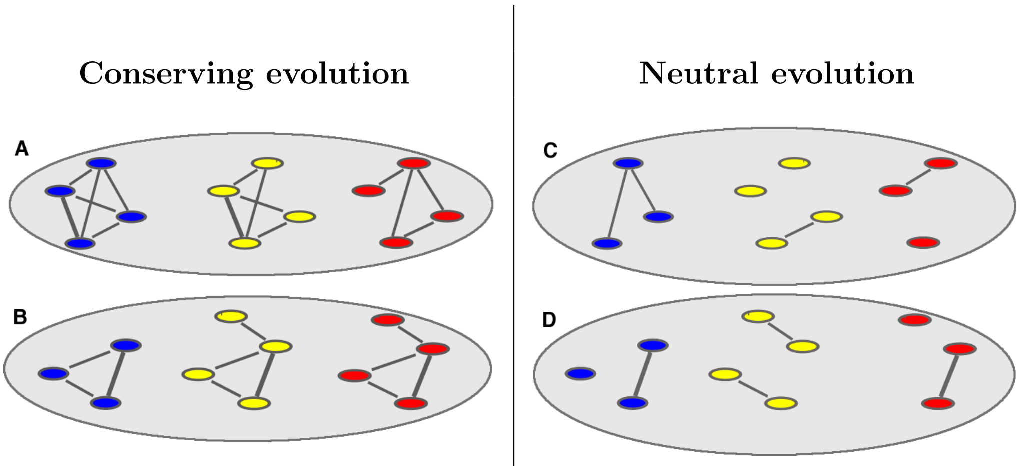
Notice that $\frac{\partial \Omega}{\partial p_e} > 0$ for $p_e < p_e^*$ and $\frac{\partial \Omega}{\partial p_e} < 0$ for $p_e > p_e^*$. Thus p_e^* is the optimal value and is selected as p_e in the new set of parameters Θ^{t+1}. Similarly we find the next value of p_1.

7.4 Distinguishing conserving and neutral evolution

We now present a proof of concept study in which we demonstrate one potential application of the parameter estimation framework. Based on available data, we try to estimate two sets of parameters for our model of network evolution. The first set of parameters will be estimated based on examples of functional modules, which should evolve under a more stringent (conserving) evolutionary scenario. The second set of parameters will be estimated based on randomly selected parts of the network. If we are successful, our two sets of parameters should be different, reflecting the difference in conservation rates among functional modules and background evolutionary rates.

Previously, we carried out all preprocessing steps ourselves using available software packages. In this study we take advantage of the TreeFam database (Li *et al.*, 2006) of protein families and phylogenies. TreeFam contains over 18000 gene family trees of which 1314 (in TreeFam A) were manually curated and are attributed greater confidence. From TreeFam A (November 2007 download) we selected 573 families which have a representative from each of the seven species studied in the previous chapter: *H. sapiens*, *M. musculus*, *R. norvegicus*, *D. melanogaster*, *C. elegans*, *S. cerevisiae*, and *A. thaliana*. Protein-protein interactions were pulled, as before, from the IntAct, DIP and MINT databases.

Our working hypothesis is that subnetworks corresponding to known functional pathways and complexes should exhibit more stringent evolutionary rates than random parts of the network. While this hypothesis is generally accepted and many positive examples were identified, it is not certain *a priori* that the difference would be sufficiently portrayed in the available data and picked up by our framework. A toy example of the experiment scheme is available in Fig. 7.1.



$$\Theta_{conserving} = \text{argmax}_{\Theta} \log P(\mathbf{x}_c|\Theta) = \text{argmax}_{\Theta} \log \sum_{\mathbf{y}_c} P(\mathbf{x}_c, \mathbf{y}_c|\Theta)$$

$$\Theta_{neutral} = \text{argmax}_{\Theta} \log P(x_r|\Theta) = \text{argmax}_{\Theta} \log \sum_{\mathbf{y}_r} P(\mathbf{x}_r, \mathbf{y}_r|\Theta)$$

Figure 7.1: *Estimating the parameters of conserving and neutral evolution of protein interaction networks. Two examples of conserved network modules for three species (blue, yellow and red) are shown on the left in (A) & (B). Two examples of subnetworks chosen at random are shown on the right in (C) & (D). Homologous proteins are aligned horizontally in each case. The model parameters for the conserving evolution scenario are derived based on the evidence for interactions within known network modules (from KEGG or MIPS). The parameters for the neutral evolution scenario are derived based on evidence for interactions within randomly chosen subnetworks. The two estimated sets of model parameters can be used to classify a new subnetwork as either conserved or evolving under the neutral scenario.*

The experiment was set up as follows. Known protein pathways were downloaded from the KEGG database (Kanehisa *et al.*, 2007). We extracted pairs of different yeast proteins which were together members of the same pathway. For each such pair of proteins, we added the corresponding pair of protein families (to which the proteins belonged to) to our set of conserved examples, but only if there was at least one protein-protein interaction observed between members of these families (in the input PPI databases). We refer to such family pairs as *conserved*. We identified 553 conserved pairs all together. Next we generated an equal in size *random* set, taking only pairs of protein families for which there was at least one interaction observed between their members. Thus both the conserved and the random set contain pairs of protein families for which some evidence for interaction between their members exist. Pairs in the conserved set fulfill an additional condition – they have at least one protein member in the same KEGG pathway. Note that in both cases a protein family can form a pair with itself.

We estimated the model parameters separately on pairs of protein families from the conserved set and on those from the random set. Note that we only estimated the parameters of the evolutionary model ($p_s, \delta_s, p_d, \delta_d, p_1$). The reliabilities of the input datasets were set as in the previous chapter (see Appendix A). Subsequently, we ran an additional experiment in which the conserved set was selected based on MIPS co-complex membership (347 family pairs altogether) instead of KEGG co-membership.

The parameter values estimated based on the three data subsets are collected in Table 7.1. The presented estimates were consistent over multiple runs of the EM procedure for different random datasets and starting from different initial parameter values. We first observe that the conservation of interactions during speciation (p_s) is greater for the conserved KEGG-based set than the random data subset. This coincides with the idea of greater conservation of essential protein interactions between species. The lesser conservation of pathway-related interactions during duplication (p_d) is perhaps less expected. One plausible explanation is that it allows duplicate pathway members to change function more easily. While the conservation of essential pathway interactions is critical, once a protein duplicates one of the copies is no longer essential. Thus it can diverge

Data	p_s	δ_s	p_d	δ_d	p_1
Random	0.85	0.02	0.59	0.001	0.999
KEGG	0.93	0.04	0.55	3.1e-6	0.999
MIPS	0.98	0.05	0.49	3.8e-5	0.999

Table 7.1: *Parameter estimates obtained for three different data subsets (Random, KEGG and MIPS) using the EM procedure.*

and gain new functionality. Essential proteins and complex members often have many interaction partners. This provides potential both for interaction loss and for formation of new interactions by adapting old interfaces. Important pathway members, which have proven useful in the cell, can thus be reused by evolution for different tasks. It is also interesting to observe that the parameters estimated on KEGG-based data are, to some degree, a mild version of the parameters estimated on MIPS data. While both sets can be used as examples of conserving evolution, the MIPS dataset clearly presents a more extreme case. This can be due to the fact that pathways correspond to a broader category of functional modules – some of them are composed of several protein complexes. Evolution may be on average more conserving for core subunits and less so for pathway interactions. Further, we notice that in all three cases the prior probability of interaction is very close to 1, suggesting that the vast majority of present interactions have their evolutionary predecessors in ancestral species. We note that the overall prior probability of interaction between ancestral proteins is expected to be much smaller (somewhere of the order of 0.001). However, here we only consider a small subspace of pairs of protein families for which at least one interaction is observed experimentally.

We consider the resulting parameter estimates reasonable, however, it is hard to rule out all possible biases in the input data and the experimental setup. For this reason any discussion of the parameter values may only have a speculative character. What is most important and encouraging for us is that there is a considerable difference between the conserved and the random models. It enables us to score the relative interaction conservation between members of arbitrary protein families.

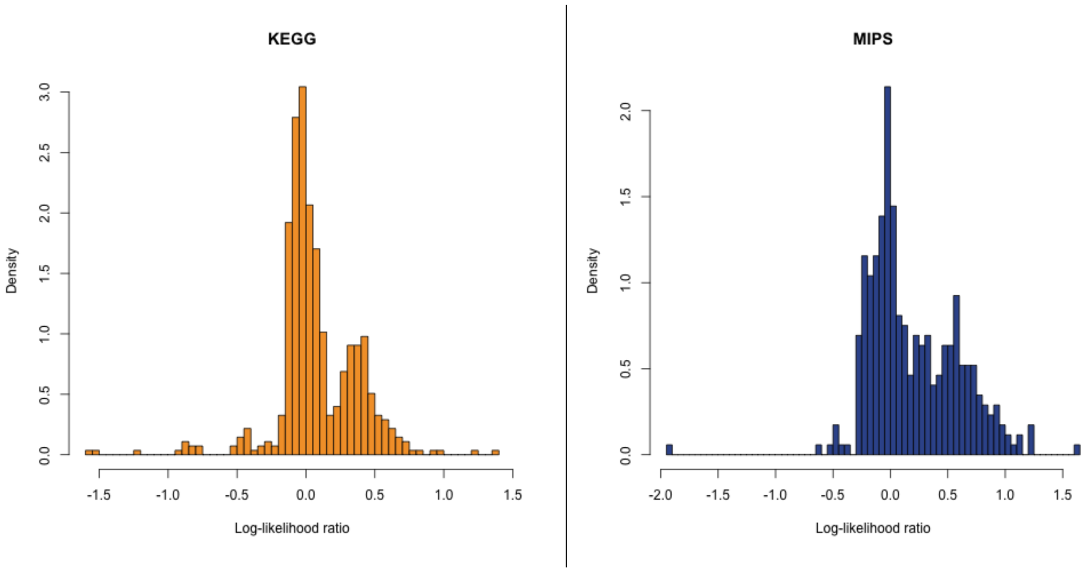


Figure 7.2: *Distribution of LLR scores among pairs of TreeFam families containing proteins belonging to common KEGG pathways (left) or MIPS complexes (right).*

For a given pair of protein families (f_1, f_2) we define the log-likelihood ratio (LLR score) as:

$$LLR(f_1, f_2) = \log \frac{P(\mathbf{x}_{f_1,f_2}|\Theta_{conserving})}{P(\mathbf{x}_{f_1,f_2}|\Theta_{neutral})},$$

where $\mathbf{x}_{f_1,f_2}$ denotes all experimental readings for interactions between the members of families f_1 and f_2. The LLR is the logarithm of the ratio of the likelihood of the data under the model of conserving evolution and under the model of neutral evolution. If $LLR > 0$ then the data are more likely under the conserving model (either based on KEGG or MIPS). We will use this score to rank arbitrary pairs of protein families. First, we analyze the scores among pairs of families used for training the conserved models.

The distribution of LLR scores for pairs of families derived based on KEGG and MIPS data are plotted in Fig. 7.2. In case of the KEGG-based pairs, the conserving model used to calculate the LLR scores was based on the KEGG database. The MIPS-based pairs were scored using the MIPS-based conserving model. Thus each pair (either KEGG-based or MIPS-based) was scored in the model for which it was used in training, and also in the neutral model. As can be expected, in both cases most protein pairs score higher under the conserving

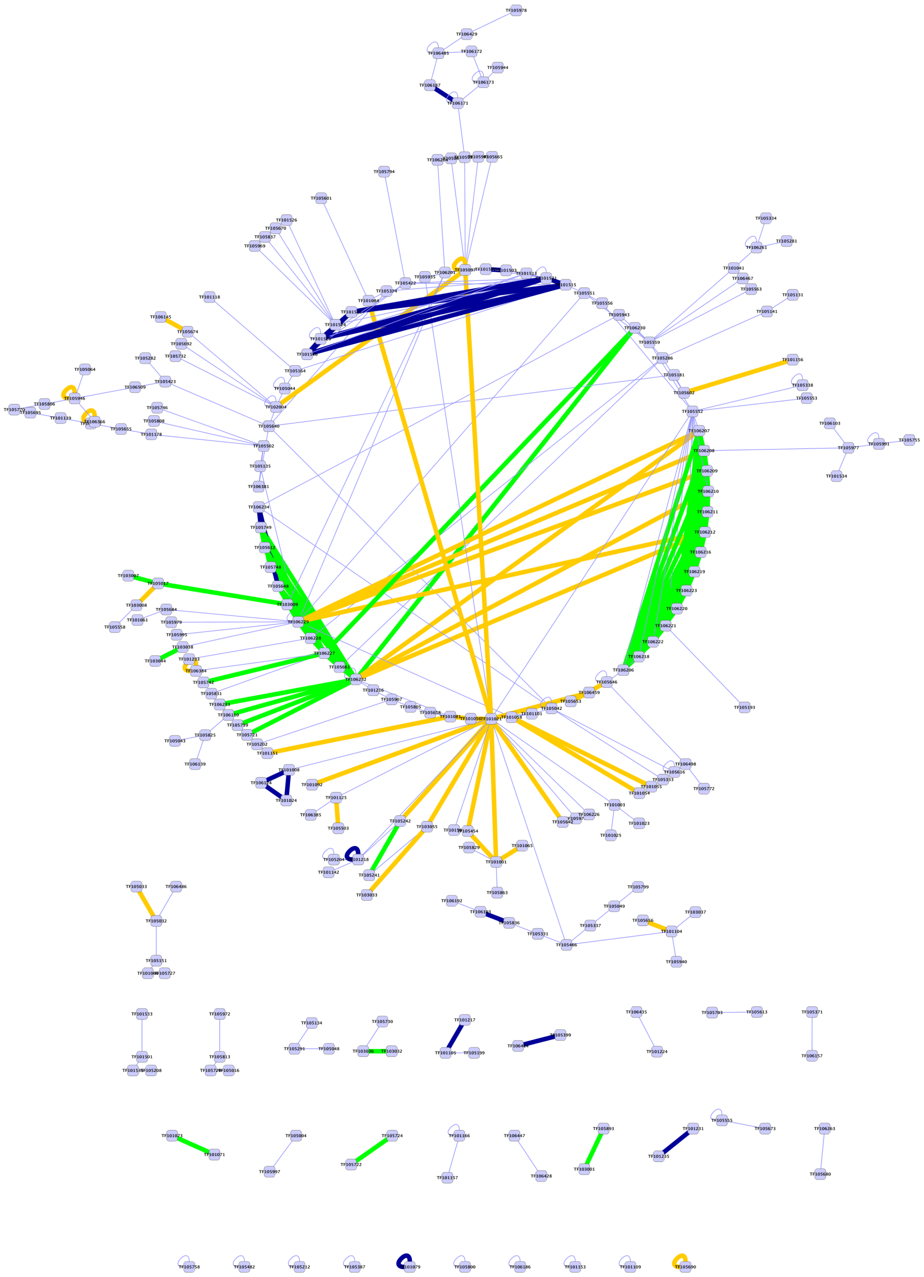


Figure 7.3: *Overview of evolutionary conserved interactions between members of TreeFam A families. Each node corresponds to a protein family. An edge denotes conservation of protein interactions between two families with LLR* > 0.4*. Thick edges correspond to known associations either based on MIPS (blue), KEGG (yellow) or both (green).*

model ($LLR > 0$). However, there is also a considerable fraction of pairs which are more likely under the neutral evolution scenario (255 KEGG and 140 MIPS) . Interestingly, the histogram for the KEGG data seems to suggest a bimodal distribution of LLR scores among the co-pathway pairs. A similar observation can be made in case of the MIPS co-complex pairs. It is possible that some of the KEGG- and MIPS-based family pairs might be significantly more conserved than others – we might be observing a mixture of two or more conservation rates. These observations provide a hint for selecting a stringent LLR threshold value, at which other highly conserved pairs can be identified.

Next we compute the LLR score for every pair of protein families using the MIPS-based conserving model and the neutral model. We construct a graph of protein families in which the edges are weighted by the LLR score. Intuitively, the edges in the graph represent the level of conservation of interactions between the members of adjacent protein families. Based on the above-described observations, we select a threshold value of 0.4 and retain only the edges in the graph with weights above this threshold. Family pairs (nodes) with no edges above the selected threshold are also discarded. Overall 251 protein families and 358 edges are retained. The resulting graph is visualized in Fig. 7.3.

The graph in Fig. 7.3 has a giant component composed of 192 nodes and 316 edges. Additionally there are 29 considerably smaller components in the network. The thick edges in the diagram correspond to protein family pairs which were previously known to be associated with functional modules and are assumed to be conserved. The 23 blue edges and 82 green edges correspond to pairs present in the MIPS-based dataset used for training (green edges were also identified in the KEGG-based dataset). Additionally 38 pairs marked by yellow edges are present only in the KEGG-based dataset and were not used for training the model in this case. We further investigate the decomposition of the giant component while raising of the LLR threshold value. When only pairs of families with LLR above 0.7 are retained, the network is comprised of 26 components – the largest of them has 18 nodes. The resulting network is presented in Fig. 7.4.

As in Chapter 5, we can assess whether the most conserved components form functionally coherent subnetworks. Overall 11 of the 26 subnetworks have a sig-

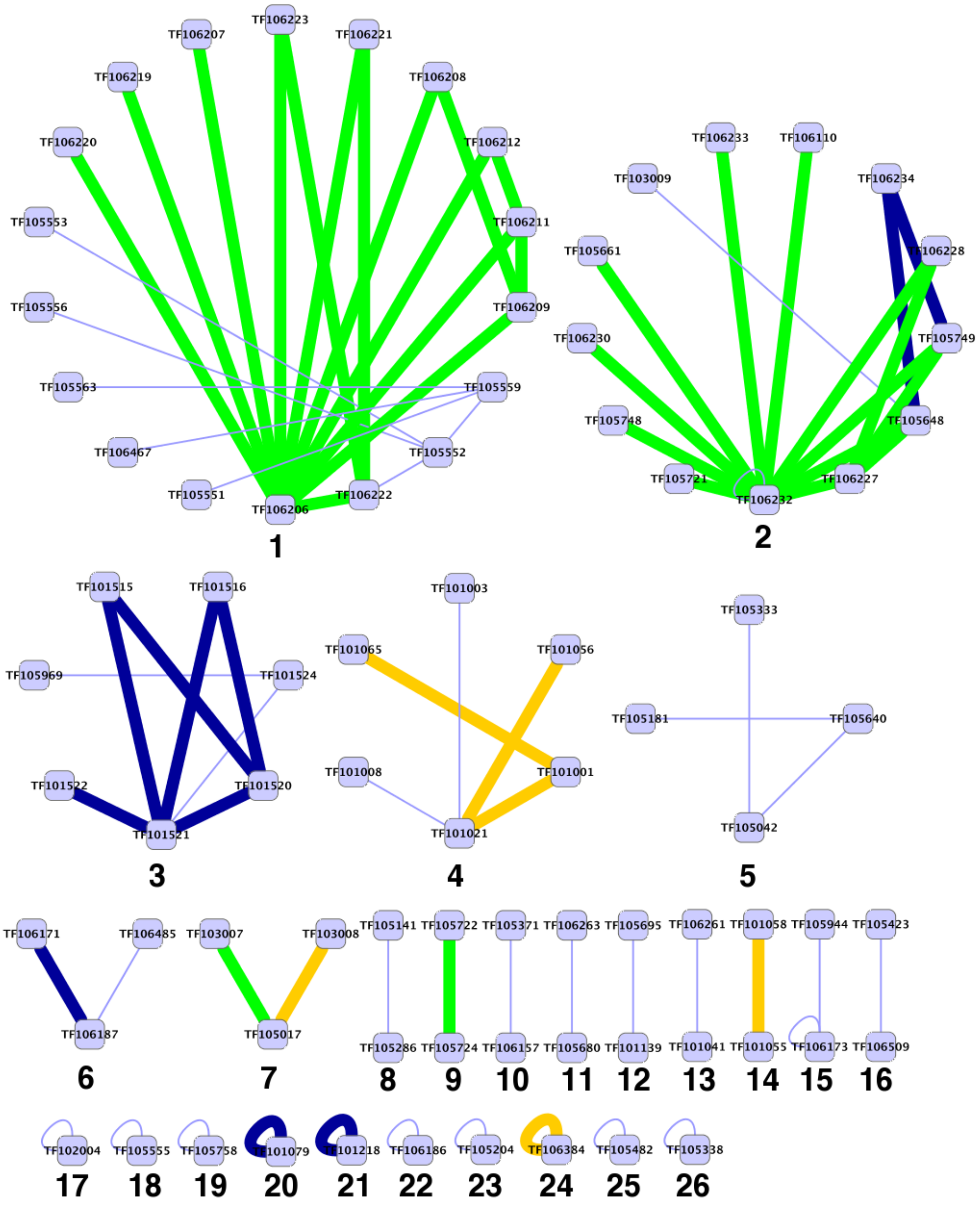


Figure 7.4: *The most conserved components of the TreeFam A family graph. An edge denotes conservation of protein interactions between two families with LLR* > 0.7. *Thick edges correspond to known associations either based on MIPS (blue), KEGG (yellow) or both (green).*

ID	Description	Arabidopsis	Yeast	Rat	Mouse	Human
1	ubiquitin-dependent protein catabolic process	<**0.0001**	<**0.0001**	<**0.0001**	<**0.0001**	<**0.0001**
2	protein catabolic process	<**0.0001**	<**0.0001**	**0.0097**	0.02	**1.0E-4**
3	translation	<**0.0001**	**1.0E-4**	0.6338	<**0.0001**	<**0.0001**
4	regulation of cell cycle	<**0.0001**	**0.0080**	**4.0E-4**	<**0.0001**	<**0.0001**
5	phosphorylation	<**0.0001**	1	0.9318	**2.0E-4**	0.01
6	regulation of transcription, DNA-dependent	<**0.0001**	**0.0035**	0.9998	**0.0014**	<**0.0001**
7	DNA replication	0.13	**0.0081**	NA	0.09	**3.0E-4**
8	protein folding	<**0.0001**	0.93	0.3797	<**0.0001**	<**0.0001**
9	DNA replication	0.1	0.23	**0.0072**	**0.0033**	0.07
10	transcription	0.05	1	1.0	1	<**0.0001**
11	RNA splicing	<**0.0001**	0.04	0.9999	**1.0E-4**	<**0.0001**
13	protein amino acid phosphorylation	0.02	1	0.0505	**0.0047**	**2.0E-4**
14	mitotic metaphase/anaphase transition	0.12	**0.0013**	NA	0.31	<**0.0001**
15	protein amino acid deacetylation	NA	0.85	NA	0.5	**3.0E-4**
16	MAPKKK cascade during cell wall biogenesis	NA	<**0.0001**	NA	NA	NA
17	phosphorylation	NA	0.74	<**0.0001**	<**0.0001**	<**0.0001**
18	mitotic spindle organization and biogenesis in nucleus	NA	**0.0022**	NA	NA	NA
19	pentose-phosphate shunt, non-oxidative branch	<**0.0001**	NA	NA	0.03	0.03
20	cell division	NA	**0.0039**	**0.0033**	<**0.0001**	<**0.0001**
21	DNA recombination	<**0.0001**	0.23	NA	0.4	0.38
22	proline biosynthetic process	0.06	0.25	<**0.0001**	<**0.0001**	<**0.0001**
23	fatty acid transport	NA	**2.0E-4**	NA	NA	0.33
25	reproduction	<**0.0001**	NA	NA	NA	NA
26	protein amino acid phosphorylation	0.62	0.98	0.9732	<**0.0001**	<**0.0001**

Table 7.2: *GO term enrichment among the connected components of protein families (found at $LLR > 0.4$).*

nificant portion of family associations supported by KEGG and/or MIPS data. Perhaps the most interesting are the yellow edges corresponding to associations present only in the KEGG-based dataset, which was not used for training. Other conserved family associations, identified *de novo* by our method, may point to unknown functional modules. To assess the potential relevance of each module, we performed a GO enrichment analysis using the Ontologizer software (Bauer *et al.*, 2008). The results are summarized in Table 7.2. Due to space considerations, p-values are reported for five of the seven species and at most one significant term is considered for each module. We find that 24 of the 26 modules are significantly enriched for a biological process term (p-value < 0.01 after correcting for multiple testing). The first two modules correspond to the proteasome complex involved in protein catabolism, which was studied in Chapter 6. Other modules are responsible for essential and highly conserved biological processes including translation, cell cycle regulation, protein folding and phosphorylation. Most of the modules have enriched annotations to the same term in multiple species. Our analysis also uncovers possibly missing annotations (denoted by NA in Table 7.2) and suggests candidate GO terms. For example, based on the evidence from four other species, Arabidopsis proteins in component number 20 might be considered for annotation to the cell division term, while rat proteins belonging to the 7-th module are suspected to take part in DNA replication.

Selecting a high threshold enables us to identify coherent and strongly conserved modules. However, we loose potentially valuable information on weaker connections. To overcome this problem and still identify closely cooperating protein families, we cluster the family graph shown in Fig. 7.3 using the affinity propagation algorithm (Frey & Dueck, 2007). The algorithm was implemented as a plug-in for the Cytoscape framework (Shannon *et al.*, 2003) by Michał Woźniak, a master's student in our group. The method identified 83 clusters, represented in different colors in Fig. 7.5, which are organized into a map of cooperating functional modules. Repeating the GO enrichment analysis for these clusters, we find that 76 of them have statistically significant biological process annotations in at least one species (p-value < 0.01 after correcting for multiple testing) and 52 have significant annotations to the same term in at least two of the seven

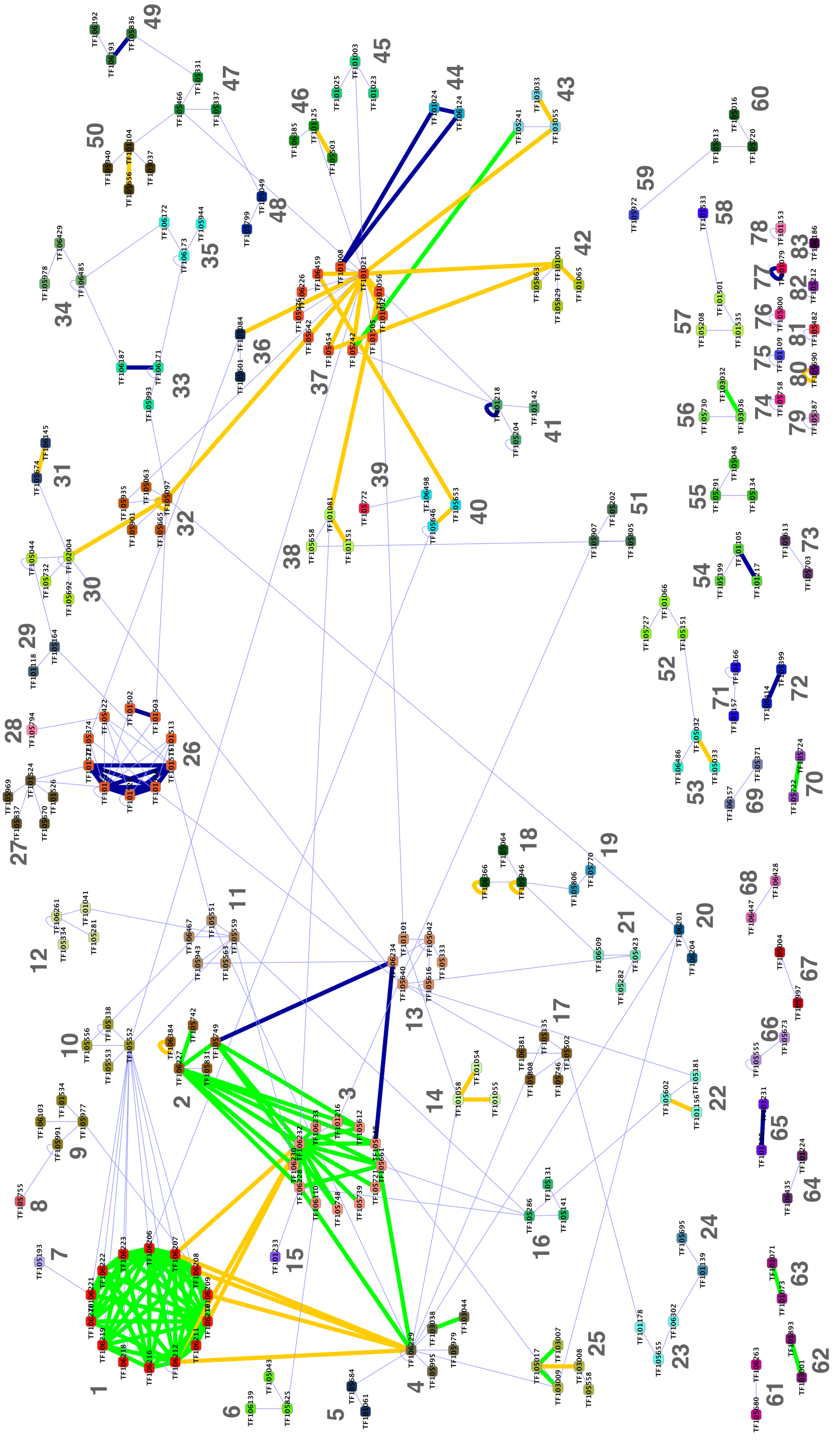


Figure 7.5: *TreeFam family graph identified at LLR threshold of 0.4 and clustered using the affinity propagation algorithm. Identified clusters are numbered and color-coded. Thick edges correspond to known associations either based on MIPS (blue), KEGG (yellow) or both (green).*

species. In Appendix B we list the annotations of 39 clusters which have significant annotations to the same term in at least two of the seven species, at a more stringent p-value < 0.001. The presented map provides a bird's-eye view of the most conserved PPI associations between members of the TreeFam A families. It can be further extended by including in the analysis the less reliable TreeFam B families. It is also possible to consider family associations with lower LLR values.

Our initial analysis was directed towards exploring the parameter space and identifying unknown conserved family associations. We used all known MIPS-based family pairs to learn the conserving model and identify novel associations which evolve as restrictively as the known functional components. An important question to ask is if our method is able to identify MIPS associations held out from training. It is also worth to investigate the robustness of our method to perturbations in the training set. To this end we applied the cross-validation procedure, iteratively learning the evolutionary parameters on 4 of 5 (approximately equal in size) partitions of the data and classifying the pairs in the partition that was held out from training. The held out portion of the data was subsequently used to validate the method's predictions. Note that each of the five partitions contained $\sim 1/5$ of the 347 MIPS-based pairs and $\sim 1/5$ of the 347 pairs selected at random. We found that our method, having only 5 parameters, does not overfit to training data. Parameters learned on subsets of the dataset (of size 4/5 of the original) were consistent with the ones learned on the entire dataset (reported in Table 7.1). As a result, we recovered 95 MIPS pairs above the 0.4 LLR threshold – each pair was identified based on the training data which did not include this particular pair. At the chosen threshold only 18 pairs from the random set were identified. Of the 95 confirmed pairs, 91 were among the 105 MIPS pairs identified by the original method trained on all available data.

Chapter 8

Conclusions

In this thesis we have developed a new framework for comparing large protein-protein interaction networks across species. The framework has the advantage of being grounded in a stochastic network growth model, which accounts for interaction conservation, loss and emergence during the course of evolution. Our approach considers the relationships between proteins and uses protein family phylogenies to guide the transfer of PPI evidence within and between species. The reconstruction of predecessor networks by our method lends considerable new insight into the PPI network evolution.

We have shown how our general framework can be applied towards the problem of identifying evolutionarily conserved subnetworks. The reconstructed network of the common ancestor of the species of interest provides an implicit alignment of the most conserved parts of the observed interactomes. We identified regions with the most probable interactions and projected them back onto the input networks. We have shown that this strategy can successfully recover known complexes and provide hypothesis about novel functional units.

We have also demonstrated how an extended version of our framework can be used to integrate and map interactions across species. Our method naturally incorporates interaction evidence from different sources and computes the posterior probability for every possible interaction. It considers the reliability of each data source and the phylogenetic relationships between protein pairs. The approach was applied to compute integrated interactomes for seven eukaryotic species, providing confidence scores for each possible edge in the network. Experimental evidence suggests that the method can accurately recover a significant part of known interactions within well-characterized protein complexes which we have studied. We have also provided many interaction predictions that await experimental verification. Interestingly, detailed analysis of CAPPI results uncovered distinct PPI profiles among homologous proteins, establishing interaction-based partitioning of large protein families.

Finally, we have derived an EM-based procedure to estimate the parameters of our model directly from data. This procedure was employed to derive rates of rewiring events within known functional modules, as well as the background rates in randomly selected subnetworks. As a result, we were able to identify two considerably different sets of parameters for conserving and neutral PPI evolution. The two parameter sets can be used to identify instances of conserved cooperation between protein families in arbitrary parts of the interactome. In a preliminary application to a number of manually-curated protein families, we were able to recover a significant number of known functional associations, as well as identify novel ones supported by GO annotations. The computed family associations were used to construct a draft version of a map of co-functioning network units. In addition to these applications, our parameter estimates provide data for theoretical considerations on the rate of network growth and divergence.

The field of biological network analysis is young and developing rapidly. New computational approaches are presented every year and advances in experimental techniques are being made. As the field matures and acquires better validation frameworks and new data, the strengths and weaknesses of each approach should become more clear. As for our method, we are also thinking of possible improvements. Enhancements that can be readily applied include deriving specific PPI conservation rates for each species, taking into account evolutionary distances. We should also be able to infer the reliability of each input dataset directly from data using the EM framework. Another good strategy, in our opinion, is to use precomputed and curated sources of protein families and phylogenies, which should make our results both more accurate and better annotated. Preliminary steps towards this goal were carried out in Chapter 7 using the TreeFam database.

We are also considering other applications of the proposed framework. As we mentioned in the introduction, biological networks are not limited to protein-

protein interactions. Other kinds of associations such as regulatory or genetic (e.g. epistasic) might also be conserved – perhaps to a smaller extent, as evidenced by recent investigations. The applicability of our framework, or other network comparison methods, to these data is yet to be evaluated. Construction of multi-level interaction maps and their subsequent analysis should result in better understanding of the architecture and functioning of cellular machinery. As new studies indicate, biological network analysis can also contribute to unraveling the mechanisms of complex diseases, including various cancer types, by providing system-level information inherently valuable for prognosis and treatment.

Appendix A

Datasets and parameter settings used for mapping seven eukaryotic interactomes

A.1 Input datasets and data preprocessing

We have downloaded the protein sequence and annotation data from the Integr8 database (Kersey *et al.*, 2005) (December, 2007 download). The input PPI data for the seven species was the same as in the InteroPorc implementation (interlog approach). The dataset included merged PPI data from the latest releases of three major databases: IntAct (Hermjakob *et al.*, 2004) (2008-08-22 version), MINT (Chatr-aryamontri *et al.*, 2007) (2008-05-16 version) and DIP (Salwinski *et al.*, 2004) (2008-07-08 version). We downloaded the input dataset for each species separately from the InteroPorc website `http://biodev.extra.cea.fr/interoporc/Default.aspx`.

Protein sequences were preprocessed leaving only the longest splice variants for each gene and clustered using the MCL algorithm (Enright *et al.*, 2002), which identified 21759 disjoint protein families. We further filtered out the families which contained sequences from less than three species leaving 4083 conserved families. Additional 10 largest families were removed due to poor sequence overlap. For each of the remaining 4073 families we constructed a phylogenetic tree and reconciled it with the species tree of the seven organisms. For the purpose of small-scale case studies we made minor corrections in six families (adding proteins missed by the automated preprocessing of the Integr8 sequence database). All other steps of the analysis were performed automatically without manual curation.

A.2 Parameter settings

The downloaded PPI data was split by species and source experiments (according to PubMed ID). Reliability parameters for large-scale input dataset used in CAPPI-Integ and CAPPI-Pred were based on the estimates from Hart *et al.* (2006), Deng *et al.* (2003) and Patil & Nakamura (2005):

First author	Year	Reliability
Gavin	2006	0.28
Giot	2003	0.2
Krogan	2006	0.27
Ewing	2007	0.7
Ito	2001	0.18
Li	2004	0.29
Gavin	2002	0.67
Ho	2002	0.27
Steltz	2005	0.15
Rual	2005	0.32
Hazbun	2003	0.31
Stanyon	2004	0.7
Formstecher	2005	0.45
Bouwmeester	2004	0.72
Uetz	2000	0.53

Smaller datasets were merged into one single dataset with reliability 0.9. During initial tests, the method showed that it was robust to variation in reliability parameters over a wide range of values. Thus no special optimization was necessary. The number of true interactions in each species was estimated as in Stumpf *et al.* (2008).

In case of CAPPI-Integ-3sp, in order to enable direct comparison with the Domain-ML approach, we set the false positive rate of each experiment to 0.0003 and the false negative rate of each experiment to 0.85, as was done by Liu *et al.* (2005).

The parameters of the model of network evolution were set (in all cases) to the following conservative values: $p_d = 0.95$, $\delta_d = 0.001$, $p_s = 0.99$ and $\delta_s = 0.001$.

A.3 Integrating datasets with different reliabilities and coverage

The conditional probabilities corresponding to true positive rate, false positive rate, false negative rate, and true negative rate of each experiment were computed as follows:

$$\begin{aligned}
Pr(X^{o_h^{(i)}}_{n_x,n_y} = 1 | X^{G_{i,m_i}}_{n_x,n_y} = 1) &= \frac{Rel(o_h^{(i)})|o_h^{(i)}|}{|E_{i,m_i}|} \\
Pr(X^{o_h^{(i)}}_{n_x,n_y} = 1 | X^{G_{i,m_i}}_{n_x,n_y} = 0) &= \frac{(1 - Rel(o_h^{(i)}))|o_h^{(i)}|}{|E'_{i,m_i}|} \\
Pr(X^{o_h^{(i)}}_{n_x,n_y} = 0 | X^{G_{i,m_i}}_{n_x,n_y} = 1) &= 1 - \frac{Rel(o_h^{(i)})|o_h^{(i)}|}{|E_{i,m_i}|} \\
Pr(X^{o_h^{(i)}}_{n_x,n_y} = 0 | X^{G_{i,m_i}}_{n_x,n_y} = 0) &= 1 - \frac{(1 - Rel(o_h^{(i)}))|o_h^{(i)}|}{|E'_{i,m_i}|}.
\end{aligned}$$

A.4 Reference datasets

The GO annotations for considered proteins and background protein populations were downloaded from the Integr8 database (December 2008 download). The functional similarity scores were computed separately for each protein pair using the SemSim Bioconductor package http://www.bioconductor.org/packages/

2.0/bioc/html/SemSim.html and averaged over the number of predicted interactions or interactions in the input datasets.

Our second kind of quality assessment was based on estimating the ratio of true positive and false positive interactions. We used separate reference datasets to determine binary and co-complex true positive PPIs. Protein pairs which where not found in the reference dataset and had differential sub-cellular localizations were counted as false positives. Below we list the reference datasets used in each case.

Yeast reference datasets A set of 3388 gold-standard yeast binary PPIs was prepared by merging the LC-multiple set from Reguly *et al.* (2006) and gold standard dataset Binary-GS from Yu *et al.* (2008), both downloaded from http://interactome.dfci.harvard.edu/S_cerevisiae/host.php?page=download. The co-complex reference dataset of 21069 protein pairs was comprised by extracting pairs of proteins from yeast complexes listed in the MIPS complex catalog of Mewes *et al.* (2006) (ftp://ftpmips.gsf.de/yeast/catalogues/complexcat/) and in the CYC2008 Complex dataset of Pu *et al.* (2009) (http://wodaklab.org/cyc2008/downloads). We discarded the complexes identified in high-throughput experiments (MIPS category 550). For CAPPI-Integ validation the complex and binary reference datasets were merged. For CAPPI-Pred experiments the binary and co-complex reference datasets were used separately and an additional dataset of experimental PPIs (binary and co-complex) was comprised from all previous reference datasets and from the left-out yeast input data, as well as a recent Y2H experiment CCSB-YI1 (Yu *et al.*, 2008) (totaling 73252 PPIs altogether). Sub-cellular localization for yeast proteins were extracted from the MIPS sub-cellular catalog. Altogether there were 4857065 differentially localized protein pairs. All datasets were most recent as of December 2008.

Human reference datasets A reference set of 36244 binary PPIs was comprised from interactions downloaded from the HPRD database (Prasad *et al.*, 2008), which stores curated interactions, mostly from small-scale studies. Co-complex pairs were extracted from HPRD complexes (9669 pairs). For CAPPI-Pred experiments the binary and co-complex reference datasets were used separately and an additional dataset (All) of experimental PPIs (binary and co-complex) was comprised from all previous reference datasets and from the left-out human input data (totaling 57093 PPIs altogether). Sub-cellular localization for human proteins were extracted from the HPRD sub-cellular catalog. Altogether there were 41647579 differentially localized protein pairs. Most recent HPRD datasets were downloaded in August 2008.

Appendix B

GO enrichment analysis for clusters of TreeFam families

Table B.1: *GO term enrichment analysis for clusters of co-functioning protein families identified with affinity propagation. Listed are clusters with significant biological process annotations to a given term in at least two species (p-value* < 0.001 *after correcting for multiple testing).*

ID	GO term	Description
1	GO:0043285	biopolymer catabolic process
1	GO:0030163	protein catabolic process
1	GO:0044267	cellular protein metabolic process
1	GO:0044265	cellular macromolecule catabolic process
1	GO:0044260	cellular macromolecule metabolic process
1	GO:0006511	ubiquitin-dependent protein catabolic process
1	GO:0044248	cellular catabolic process
1	GO:0006508	proteolysis
1	GO:0051603	proteolysis involved in cellular protein catabolic process
1	GO:0019941	modification-dependent protein catabolic process
1	GO:0044257	cellular protein catabolic process
1	GO:0043632	modification-dependent macromolecule catabolic process
1	GO:0009056	catabolic process

Continued on next page

Table B.1 – continued from previous page

ID	GO term	Description
1	GO:0009057	macromolecule catabolic process
1	GO:0008152	metabolic process
1	GO:0009987	cellular process
1	GO:0043283	biopolymer metabolic process
1	GO:0019538	protein metabolic process
1	GO:0043170	macromolecule metabolic process
1	GO:0044238	primary metabolic process
1	GO:0044237	cellular metabolic process
3	GO:0019538	protein metabolic process
3	GO:0044267	cellular protein metabolic process
3	GO:0044260	cellular macromolecule metabolic process
3	GO:0006508	proteolysis
3	GO:0043285	biopolymer catabolic process
3	GO:0030163	protein catabolic process
3	GO:0009056	catabolic process
3	GO:0009057	macromolecule catabolic process
7	GO:0042493	response to drug
10	GO:0007165	signal transduction
10	GO:0007154	cell communication
11	GO:0016311	dephosphorylation
12	GO:0006464	protein modification process
12	GO:0043283	biopolymer metabolic process
12	GO:0000398	nuclear mRNA splicing, via spliceosome
12	GO:0043170	macromolecule metabolic process
12	GO:0000375	RNA splicing, via transesterification reactions
12	GO:0008380	RNA splicing
12	GO:0000377	RNA splicing, via transesterification reactions with bulged adenosine as nucleophile
12	GO:0044238	primary metabolic process
12	GO:0044237	cellular metabolic process
12	GO:0006397	mRNA processing
12	GO:0016310	phosphorylation
12	GO:0043687	post-translational protein modification
12	GO:0006468	protein amino acid phosphorylation
12	GO:0006796	phosphate metabolic process
12	GO:0006793	phosphorus metabolic process
13	GO:0007283	spermatogenesis
13	GO:0048232	male gamete generation

Continued on next page

Table B.1 – continued from previous page

ID	GO term	Description
14	GO:0007091	mitotic metaphase/anaphase transition
16	GO:0006457	protein folding
17	GO:0000165	MAPKKK cascade
17	GO:0045454	cell redox homeostasis
17	GO:0045859	regulation of protein kinase activity
17	GO:0051347	positive regulation of transferase activity
17	GO:0050790	regulation of catalytic activity
17	GO:0051338	regulation of transferase activity
17	GO:0043406	positive regulation of MAPK activity
17	GO:0043405	regulation of MAPK activity
17	GO:0043549	regulation of kinase activity
17	GO:0065009	regulation of a molecular function
17	GO:0043085	positive regulation of enzyme activity
17	GO:0045860	positive regulation of protein kinase activity
17	GO:0000187	activation of MAPK activity
18	GO:0009161	ribonucleoside monophosphate metabolic process
18	GO:0009156	ribonucleoside monophosphate biosynthetic process
18	GO:0009260	ribonucleotide biosynthetic process
18	GO:0009123	nucleoside monophosphate metabolic process
18	GO:0009259	ribonucleotide metabolic process
18	GO:0009124	nucleoside monophosphate biosynthetic process
18	GO:0009116	nucleoside metabolic process
22	GO:0007059	chromosome segregation
23	GO:0051169	nuclear transport
23	GO:0008104	protein localization
23	GO:0033036	macromolecule localization
23	GO:0006606	protein import into nucleus
23	GO:0006605	protein targeting
23	GO:0045184	establishment of protein localization
23	GO:0006886	intracellular protein transport
23	GO:0046907	intracellular transport
23	GO:0006913	nucleocytoplasmic transport
23	GO:0017038	protein import
23	GO:0051170	nuclear import
23	GO:0051649	establishment of cellular localization
23	GO:0051641	cellular localization
23	GO:0015031	protein transport
23	GO:0016043	cellular component organization and biogenesis

Continued on next page

Table B.1 – continued from previous page

ID	GO term	Description
23	GO:0006810	transport
23	GO:0051234	establishment of localization
23	GO:0051179	localization
26	GO:0006413	translational initiation
26	GO:0022618	protein-RNA complex assembly
26	GO:0022613	ribonucleoprotein complex biogenesis and assembly
26	GO:0022607	cellular component assembly
26	GO:0065003	macromolecular complex assembly
26	GO:0009058	biosynthetic process
26	GO:0006412	translation
26	GO:0009059	macromolecule biosynthetic process
27	GO:0006413	translational initiation
29	GO:0006457	protein folding
30	GO:0006468	protein amino acid phosphorylation
30	GO:0044267	cellular protein metabolic process
30	GO:0016310	phosphorylation
30	GO:0007165	signal transduction
30	GO:0007154	cell communication
30	GO:0019538	protein metabolic process
30	GO:0043412	biopolymer modification
30	GO:0043687	post-translational protein modification
30	GO:0006464	protein modification process
30	GO:0006796	phosphate metabolic process
30	GO:0006793	phosphorus metabolic process
30	GO:0044260	cellular macromolecule metabolic process
30	GO:0043170	macromolecule metabolic process
30	GO:0007242	intracellular signaling cascade
31	GO:0019320	hexose catabolic process
31	GO:0006007	glucose catabolic process
31	GO:0046365	monosaccharide catabolic process
31	GO:0006096	glycolysis
32	GO:0016310	phosphorylation
33	GO:0007001	chromosome organization and biogenesis (sensu Eukaryota)
33	GO:0016569	covalent chromatin modification
33	GO:0016570	histone modification
33	GO:0051276	chromosome organization and biogenesis
33	GO:0016575	histone deacetylation
33	GO:0006355	regulation of transcription, DNA-dependent

Continued on next page

Table B.1 – continued from previous page

ID	GO term	Description
33	GO:0032774	RNA biosynthetic process
33	GO:0006351	transcription, DNA-dependent
33	GO:0006350	transcription
33	GO:0045449	regulation of transcription
33	GO:0006476	protein amino acid deacetylation
33	GO:0006323	DNA packaging
33	GO:0006325	establishment and/or maintenance of chromatin architecture
33	GO:0016568	chromatin modification
34	GO:0006259	DNA metabolic process
34	GO:0007001	chromosome organization and biogenesis (sensu Eukaryota)
34	GO:0016568	chromatin modification
34	GO:0051276	chromosome organization and biogenesis
34	GO:0006325	establishment and/or maintenance of chromatin architecture
34	GO:0006323	DNA packaging
35	GO:0016575	histone deacetylation
35	GO:0016570	histone modification
35	GO:0016569	covalent chromatin modification
35	GO:0006476	protein amino acid deacetylation
35	GO:0016568	chromatin modification
37	GO:0051301	cell division
37	GO:0000087	M phase of mitotic cell cycle
37	GO:0043283	biopolymer metabolic process
37	GO:0007067	mitosis
37	GO:0000279	M phase
37	GO:0044238	primary metabolic process
37	GO:0000278	mitotic cell cycle
37	GO:0007049	cell cycle
37	GO:0022403	cell cycle phase
37	GO:0022402	cell cycle process
42	GO:0000074	regulation of progression through cell cycle
42	GO:0051726	regulation of cell cycle
42	GO:0022403	cell cycle phase
42	GO:0000279	M phase
42	GO:0022402	cell cycle process
42	GO:0051301	cell division
42	GO:0007049	cell cycle
45	GO:0051301	cell division
46	GO:0006512	ubiquitin cycle

Continued on next page

Table B.1 – continued from previous page

ID	GO term	Description
47	GO:0043412	biopolymer modification
47	GO:0006464	protein modification process
47	GO:0043687	post-translational protein modification
47	GO:0006793	phosphorus metabolic process
47	GO:0006796	phosphate metabolic process
47	GO:0016310	phosphorylation
47	GO:0006468	protein amino acid phosphorylation
48	GO:0006950	response to stress
48	GO:0050896	response to stimulus
49	GO:0006886	intracellular protein transport
49	GO:0045039	protein import into mitochondrial inner membrane
49	GO:0007007	inner mitochondrial membrane organization and biogenesis
49	GO:0006626	protein targeting to mitochondrion
49	GO:0043681	protein import into mitochondrion
49	GO:0065002	intracellular protein transport across a membrane
49	GO:0007006	mitochondrial membrane organization and biogenesis
49	GO:0007005	mitochondrion organization and biogenesis
55	GO:0006457	protein folding
57	GO:0022618	protein-RNA complex assembly
57	GO:0022613	ribonucleoprotein complex biogenesis and assembly
57	GO:0006413	translational initiation
61	GO:0016071	mRNA metabolic process
61	GO:0006397	mRNA processing
61	GO:0006396	RNA processing
61	GO:0008380	RNA splicing
62	GO:0006261	DNA-dependent DNA replication
65	GO:0006289	nucleotide-excision repair
65	GO:0000737	DNA catabolic process, endonucleolytic
68	GO:0007031	peroxisome organization and biogenesis
71	GO:0007001	chromosome organization and biogenesis (sensu Eukaryota)
71	GO:0051276	chromosome organization and biogenesis
75	GO:0007264	small GTPase mediated signal transduction
75	GO:0007242	intracellular signaling cascade
75	GO:0007165	signal transduction
75	GO:0000902	cell morphogenesis
75	GO:0030030	cell projection organization and biogenesis
75	GO:0030036	actin cytoskeleton organization and biogenesis
75	GO:0048858	cell projection morphogenesis

Continued on next page

Table B.1 – continued from previous page

ID	GO term	Description
75	GO:0048856	anatomical structure development
75	GO:0030029	actin filament-based process
75	GO:0032989	cellular structure morphogenesis
75	GO:0009653	anatomical structure morphogenesis
75	GO:0007010	cytoskeleton organization and biogenesis
75	GO:0032990	cell part morphogenesis
75	GO:0030031	cell projection biogenesis
77	GO:0051301	cell division
77	GO:0007049	cell cycle
83	GO:0006561	proline biosynthetic process
83	GO:0006560	proline metabolic process
83	GO:0009084	glutamine family amino acid biosynthetic process
83	GO:0009064	glutamine family amino acid metabolic process
83	GO:0006520	amino acid metabolic process
83	GO:0008652	amino acid biosynthetic process
83	GO:0009309	amine biosynthetic process
83	GO:0044271	nitrogen compound biosynthetic process

References

Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, **215**, 403–410. 18

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M. & Sherlock, G. (2000). Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature Genetics*, **25**, 25–29. 67, 76

Bader, G. & Hogue, C. (2002). Analyzing yeast protein-protein interaction data obtained from different sources. *Nature Biotechnology*, **20**, 991–997. 5

Bader, J., Chaudhuri, A., Rothberg, J. & Chant, J. (2004). Gaining confidence in high-throughput protein interaction networks. *Nature Biotechnology*, **22**, 78–85. 11

Barabasi, A.L. & Albert, R. (1999). Emergence of scaling in random networks. *Science*, **286**, 509–512. 30, 34

Bauer, S., Grossmann, S., Vingron, M. & Robinson, P.N.N. (2008). Ontologizer 2.0 - a multifunctional tool for GO term enrichment analysis and data exploration. *Bioinformatics*, **24**, 1650–1651. 121

Baum, L.E., Petrie, T., Soules, G. & Weiss, N. (1970). A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, **41**, 164–171. 107

Bebek, G., Berenbrink, P., Cooper, C., Friedetzky, T., Nadeau, J.H. & Sahinalp, S.C. (2005). Improved duplication models for proteome network evolution. In E. Eskin, T. Ideker, B.J. Raphael & C.T. Workman, eds., *Recomb Systems Biology and Regulatory Genomics*, vol. 4023 of *Lecture Notes in Computer Science*, 119–137, Springer. 33

Bebek, G., Berenbrink, P., Cooper, C., Friedetzky, T., Nadeau, J. & Sahinalp, S.C. (2006). The degree distribution of the generalized duplication model. *Theoretical Computer Science*, **369**, 239–249. 33

Beltrao, P. & Serrano, L. (2007). Specificity and evolvability in eukaryotic protein interaction networks. *PLoS Computational Biology*, **3**, e25. 12, 86

Berg, J. & Lassig, M. (2006a). Bayesian analysis of biological networks: clusters, motifs, cross-species correlations. 14

Berg, J. & Lassig, M. (2006b). Cross-species analysis of biological networks by Bayesian alignment. *Proc Natl Acad Sci U S A*, **103**, 10967–10972. 9, 10

Bertsekas, D.P. & Tsitsiklis, J.N. (2008). *Introduction to Probability (Second edition)*. Athena Scientific. 104, 106

Bezhani, S., Winter, C., Hershman, S., Wagner, J.D., Kennedy, J.F., Kwon, C.S., Pfluger, J., Su, Y. & Wagner, D. (2007). Unique, shared, and redundant roles for the Arabidopsis SWI/SNF chromatin remodeling ATPases BRAHMA and SPLAYED. *The Plant Cell*, **19**, 403–416. 77

Bleakley, K., Biau, G. & Vert, J.P. (2007). Supervised reconstruction of biological networks with local models. *Bioinformatics*, **23**, 57–65. 11

Bollobás, B. (2001). *Random Graphs*. Cambridge University Press. 29, 30

Bollobás, B. (2003). Mathematical results on scale-free random graphs. In *In Handbook of Graphs and Networks*, 1–34, Wiley-VCH. 29, 30, 31, 32, 34

Bollobás, B. & Riordan, O. (2004). The diameter of a scale-free random graph. *Combinatorica*, **24**, 5–34. 31, 32

Bollobás, B., Riordan, O., Spencer, J. & Tusnády, G. (2001). The degree sequence of a scale-free random graph process. *Random Structures and Algorithms*, **18**, 279–290. 31

Bork, P., Jensen, L., von Mering, C., Ramani, A., Lee, I. & Marcotte, E. (2004). Protein interaction networks from yeast to human. *Current Opinion in Structural Biology*, **14**, 292–299. 12

Brin, S. & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*, 107–117. 9

Brohee, S. & van Helden, J. (2006). Evaluation of clustering algorithms for protein-protein interaction networks. *BMC Bioinformatics*, **7**, 488. 20

Buneman, P. (1971). The recovery of trees from measures of dissimilarity. In *Mathematics in the Archaeological and Historical Sciences, F. R. Hodson, D. G. Kendall and P. Tautu (Eds)*, 387–395, Edinburgh University Press. 24

Cagney, G., Uetz, P. & Fields, S. (2001). Two-hybrid analysis of the Saccharomyces cerevisiae 26S proteasome. *Physiological Genomics*, **7**, 27–34. 77

Chagoyen, M., Carazo, J. & Montano, A.P. (2008). Assessment of protein set coherence using functional annotations. *BMC Bioinformatics*, **9**, 444. 67

Chatr-aryamontri, A., Ceol, A., Palazzi, L.M., Nardelli, G., Schneider, M.V., Castagnoli, L. & Cesareni, G. (2007). MINT: the Molecular INTeraction database. *Nucleic Acids Research*, **35**, D572–D574. 74, 127

Chen, C., Huang, C., Chen, S., Liang, J., Lin, W., Ke, G., Zhang, H., Wang, B., Huang, J., Han, Z., Ma, L., Huo, K., Yang, X., Yang, P., He, F. & Tao, T. (2008). Subunit-subunit interactions in the human 26S proteasome. *Proteomics*, **8**, 508–520. 77, 92

Chen, K. & Durand, D. (2000). Notung: A program for dating gene duplications and optimizing gene family trees. *Journal of Computational Biology*, **7**, 429–447. 26

Chen, X.W. & Liu, M. (2005). Prediction of protein-protein interactions using random decision forest framework. *Bioinformatics*, **21**, 4394–4400. 11

Cooper, G.F. (1990). The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, **42**, 393–405. 41

Cusick, M., Klitgord, N., Vidal, M. & Hill, D. (2005). Interactome: gateway into systems biology. *Human Molecular Genetics*, **14**, 171–181. 2

David, P.S., Tanveer, R. & Port, J.D. (2007). FRET-detectable interactions between the ARE binding proteins, HuR and p37AUF1. *RNA*, **13**, 1453–1468. 98

Dempster, A.P., Laird, N.M. & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, **39**, 1–38. 103, 107

Deng, M., Mehta, S., Sun, F. & Chen, T. (2002). Inferring domain-domain interactions from protein-protein interactions. *Genome Research*, **12**, 1540–1548. 11

Deng, M., Sun, F. & Chen, T. (2003). Assessment of the reliability of protein-protein interactions and protein function prediction. In *Proc. Eighth Pacific Symposium on Biocomputing*, 140–151. 128

Dujon, B. & Sherman, D. (2004). Genome evolution in yeasts. *Nature*, **430**, 35–44. 20

Durand, D., Halldorsson, B. & Vernot, B. (2006). A hybrid micro-macroevolutionary approach to gene tree reconstruction. *Journal of Computational Biology*, **13**, 320–335. 50

DURBIN, R., EDDY, S.R., KROGH, A. & MITCHISON, G. (1998). *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press. 16, 17, 18, 20, 23, 107

DUTKOWSKI, J. & TIURYN, J. (2007). Identification of functional modules from conserved ancestral protein-protein interactions. *Bioinformatics*, **23**, i149–i158. 10, 14, 49, 68

EDGAR, R.C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, **32**, 1792–1797. 20

ENRIGHT, A., ILIOPOULOS, I., KYRPIDES, N. & OUZOUNIS, C. (1999). Protein interaction maps for complete genomes based on gene fusion events. *Nature*, **402**, 86–90. 10

ENRIGHT, A.J., VAN DONGEN, S. & OUZOUNIS, C.A. (2002). An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, **30**, 1575–1584. 20, 21, 50, 127

ERDÖS, P. & RÉNYI, A. (1959). On random graphs. *Publicationes Mathematicae Debrecen*, **6**, 290–297. 30

FARRONA, S., HURTADO, L., BOWMAN, J.L. & REYES, J.C. (2004). The Arabidopsis thaliana SNF2 homolog AtBRM controls shoot development and flowering. *Development*, **131**, 4965–4975. 77

FELSENSTEIN, J. (2005). *PHYLIP (Phylogeny Inference Package) version 3.6*. Distributed by the author. Department of Genome Sciences, University of Washington, Seattle, http://evolution.genetics.washington.edu/phylip.html. 50

FLANNICK, J., NOVAK, A., SRINIVASAN, B.S., MCADAMS, H.H. & BATZOGLOU, S. (2006). Graemlin: general and robust alignment of multiple large interaction networks. *Genome Research*, **16**, 1169–1181. 7, 10, 50, 66

Flannick, J., Novak, A., Do, C.B., Srinivasan, B.S. & Batzoglou, S. (2008). Automatic parameter learning for multiple network alignment. In M. Vingron & L. Wong, eds., *RECOMB*, vol. 4955 of *Lecture Notes in Computer Science*, 214–231, Springer. 9

Frey, B.J.J. & Dueck, D. (2007). Clustering by passing messages between data points. *Science*, **15**, 972–976. 121

Gavin, A.C., Aloy, P., Grandi, P., Krause, R., Boesche, M., Marzioch, M., Rau, C., Jensen, L.J., Bastuck, S., Dümpelfeld, B., Edelmann, A., Heurtier, M.A., Hoffman, V., Hoefert, C., Klein, K., Hudak, M., Michon, A.M., Schelder, M., Schirle, M., Remor, M., Rudi, T., Hooper, S., Bauer, A., Bouwmeester, T., Casari, G., Drewes, G., Neubauer, G., Rick, J.M., Kuster, B., Bork, P., Russell, R.B. & Superti-Furga, G. (2006). Proteome survey reveals modularity of the yeast cell machinery. *Nature*, **440**, 631–636. 4

Gilbert, E.N. (1959). Random graphs. *Annals of Mathematical Statistics*, **30**, 1141–1144. 29

Giot, C., L, Bader, J.C., Brouwer, C.C., Chaudhuri, A.C., Kuang, B.C., Li, Y.C., Hao, Y.C., Ooi, C.C., Godwin, B.C., Vitols, E.C., Vijayadamodar, G.C., Pochart, P.C., Machineni, H.C., Welsh, M.C., Kong, Y.C., Zerhusen, B.C., Malcolm, R.C., Varrone, Z.C., Collis, A.C., Minto, M.C., Burgess, S.C., McDaniel, L.C., Stimpson, E.C., Spriggs, F.C., Williams, J.C., Neurath, K.C., Ioime, N.C., Agee, M.C., Voss, E.C., Furtak, K.C., Renzulli, R.C., Aanensen, N.C., Carrolla, S.C., Bickelhaupt, E.C., Lazovatsky, Y.C., Dasilva, A.C., Zhong, J.C., Stanyon, C.C., Finley, J.R.C., White, K.C., Braverman, M.C., Jarvie, T.C., Gold, S.C., Leach, M.C., Knight, J.C., Shimkets, R.C., McKenna, M.C., Chant, J.C. & Rothberg, J. (2003). A protein interaction map of Drosophila melanogaster. *Science*, **302**, 1727–1736. 74

Glickman, M.H. & Ciechanover, A. (2002). The ubiquitin-proteasome proteolytic pathway: destruction for the sake of construction. *Physiol Rev*, **82**, 373–428. 89

Górecki, P. & Tiuryn, J. (2006a). DLS-trees: a model of evolutionary scenarios. *Theoretical Computer Science*, **359**, 378–399. 24, 25, 26

Górecki, P. & Tiuryn, J. (2006b). Inferring phylogeny from whole genomes. *Bioinformatics*, **23**, e116–e122. 26

Groll, M. & Huber, R. (2005). Purification, crystallization, and X-ray analysis of the yeast 20S proteasome. *Methods in Enzymology*, **398**, 329–336. 89

Hart, T.G., Ramani, A.K. & Marcotte, E.M. (2006). How complete are current yeast and human protein-interaction networks? *Genome Biology*, **7**, 120. 5, 81, 128

Hermjakob, H., Montecchi-Palazzi, L., Lewington, C., Mudali, S., Kerrien, S., Orchard, S., Vingron, M., Roechert, B., Roepstorff, P., Valencia, A., Margalit, H., Armstrong, J., Bairoch, A., Cesareni, G., Sherman, D. & Apweiler, R. (2004). IntAct: an open source molecular interaction database. *Nucleic Acids Reseach*, **32**, D452–D455. 74, 127

Higgins, D., Thompson, J.D., Higgins, D.G. & Gibson, T.J. (1994). CLUSTALW: 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. 19, 20, 50

Hirsh, E. & Sharan, R. (2006). Identification of conserved protein complexes based on a model of protein network evolution. *Bioinformatics*, **23**, e170–e176. 9, 10

Hormozdiari, F., Berenbrink, P., Przulj, N. & Sahinalp, S.C.C. (2007). Not all scale-free networks are born equal: The role of the seed graph in PPI network evolution. *PLoS Computational Biology*, **3**, e118. 34

HURLEY, J.H. & EMR, S.D. (2006). The ESCRT complexes: structure and mechanism of a membrane-trafficking network. *Annual Review of Biophysics and Biomolecular Structure*, **35**, 277–298. 77, 94

HURTADO, L., FARRONA, S. & REYES, J.C. (2006). The putative SWI/SNF complex subunit BRAHMA activates flower homeotic genes in Arabidopsis thaliana. *Plant Molecular Biology*, **62**, 291–304. 77

IDEKER, T. & SHARAN, R. (2008). Protein networks in disease. *Genome Research*, **18**, 644–652. 6

IDEKER, T., GALITSKI, T. & HOOD, L. (2001). A new approach to decoding life: systems biology. *Annual Review of Genomics and Human Genetics*, **2**, 343–372. 2

INTERNATIONAL HUMAN GENOME SEQUENCING CONSORTIUM (2001). Initial sequencing and analysis of the human genome. *Nature*, **409**, 860–921. 1

ISPOLATOV, I., KRAPIVSKY, P.L., MAZO, I. & YURYEV, A. (2005a). Cliques and duplication-divergence network growth. *New Journal of Physics*, **7**, 145. 34

ISPOLATOV, I., KRAPIVSKY, P.L. & YURYEV, A. (2005b). Duplication-divergence model of protein interaction network. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, **71**, 061911. 33

ITO, T., CHIBA, T., OZAWA, R., YOSHIDA, M., HATTORI, M. & SAKAKI, Y. (2001). A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc Natl Acad Sci U S A*, **98**, 4569–4574. 2, 5, 74

JANSEN, R., YU, H., GREENBAUM, D., KLUGER, Y., KROGAN, N., CHUNG, S., EMILI, A., SNYDER, M., GREENBLATT, J. & GERSTEIN, M. (2003). A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, **302**, 449–453. 11, 77

JERZMANOWSKI, A. (2007). SWI/SNF chromatin remodeling and linker histones in plants. *Biochimica et Biophysica Acta*, **1769**, 330–345. 98, 100, 102

Jothi, R., Kann, M. & Przytycka, T. (2005). Predicting protein-protein interaction by searching evolutionary tree automorphism space. *Bioinformatics*, **21**, 241–250. 11

Juan, D., Pazos, F. & Valencia, A. (2008). High-confidence prediction of global interactomes based on genome-wide coevolutionary networks. *Proc Natl Acad Sci U S A*, **105**, 934–939. 11

Just, W. (2001). Computational complexity of multiple sequence alignment with SP-score. *Journal of Computational Biology*, **8**, 615–623. 19

Kalaev, M., Bafna, V. & Sharan, R. (2008). Fast and accurate alignment of multiple protein networks. In M. Vingron & L. Wong, eds., *RECOMB*, vol. 4955 of *Lecture Notes in Computer Science*, 246–256, Springer. 9

Kanehisa, M., Araki, M., Goto, S., Hattori, M., Hirakawa, M., Itoh, M., Katayama, T., Kawashima, S., Okuda, S., Tokimatsu, T. & Yamanishi, Y. (2007). KEGG for linking genomes to life and the environment. *Nucleic Acids Research*, D480–D484. 114

Kelley, B.P., Sharan, R., Karp, R.M., Sittler, T., Root, D.E., Stockwell, B.R. & Ideker, T. (2003). Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proc Natl Acad Sci U S A*, **100**, 11394–11399. 7, 58

Kersey, P., Bower, L., Morris, L., Horne, A., Petryszak, R., Kanz, C., Kanapin, E., Das, U., Michoud, K., Phan, I., Gattiker, R., Kulikova, T., Faruque, N., Duggan, K., Mclaren, P., Reimholz, B., Duret, L., Penel, S., Reuter, I. & Apweiler, R. (2005). Integr8 and Genome Reviews: integrated views of complete genomes and proteomes. *Nucleic Acids Research*, **33**, 297–302. 74, 127

Kondrashov, F., Rogozin, I., Wolf, Y. & Koonin, E. (2002). Selection in the evolution of gene duplications. *Genome Biology*, **2**, RESEARCH0008. 34

Koyuturk, M., Kim, Y., Topkara, U., Subramaniam, S., Szpankowski, W. & Grama, A. (2006). Pairwise alignment of protein interaction networks. *Journal of Computational Biology*, **13**, 182–199. 7, 10, 58

Krogan, N.J., Cagney, G., Yu, H., Zhong, G., Guo, X., Ignatchenko, A., Li, J., Pu, S., Datta, N., Tikuisis, A.P., Punna, T., Peregran-Alvarez, J.a.M., Shales, M., Zhang, X., Davey, M., Robinson, M.D., Paccanaro, A., Bray, J.E., Sheung, A., Beattie, B., Richards, D.P., Canadien, V., Lalev, A., Mena, F., Wong, P., Starostine, A., Canete, M.M., Vlasblom, J., Wu, S., Orsi, C., Collins, S.R., Chandran, S., Haw, R., Rilstone, J.J., Gandi, K., Thompson, N.J., Musso, G., St Onge, P., Ghanny, S., Lam, M.H.Y., Butland, G., Altaf-Ul, A.M., Kanaya, S., Shilatifard, A., O'Shea, E., Weissman, J.S., Ingles, J.C., Hughes, T.R., Parkinson, J., Gerstein, M., Wodak, S.J., Emili, A. & Greenblatt, J.F. (2006). Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. *Nature*, **440**, 637–643. 4

Lauritzen, S.L. (1995). The EM algorithm for graphical association models with missing data. *Computational Statistics & Data Analysis*, **19**, 191–201. 107

Lee, H., Deng, M., Sun, F. & Chen, T. (2006). An integrated approach to the prediction of domain-domain interactions. *BMC Bioinformatics*, **7**, 269. 11, 12

Lehner, B. & Sanderson, C.M. (2004). A protein interaction framework for human mRNA degradation. *Genome Research*, **14**, 1315–1323. 77, 96, 98

Li, C., Siming, Armstrong, C.C., Bertin, N.C., Ge, H.C., Milstein, S.C., Boxem, M.C., Vidalain, P.O.C., Han, J.D.C., Chesneau, A.C., Hao, T.C., Goldberg, D.C., Li, N.C., Martinez, M.C., Rual, J.F.C., Lamesch, P.C., Xu, L.C., Tewari, M.C., Wong, S.C., Zhang, L.C., Berriz, G.C., Jacotot, L.C., Vaglio, P.C., Reboul, J.C., Hirozane-Kishikawa, T.C., Li, Q.C., Gabel, H.C., Elewa, A.C., Baumgartner, B.C., Rose, D.C., Yu, H.C., Bosak, S.C., Sequerra, R.C., Fraser, A.C., Mango, S.C., Saxton, W.C., Strome, S.C., van den Heuvel, S.C., Piano, F.C., Vandenhaute, J.C., Sardet, C.C., Gerstein, M.C., Doucette-Stamm, L.C., Gunsalus, K.C., Harper, J.C., Cusick, M.C., Roth, F.C., Hill, D.C. & Vidal, M. (2004). A map of the interactome network of the metazoan C. elegans. *Science*, **303**, 540–543. 74

Li, H., Coghlan, A., Ruan, J., Coin, L.J., Heriche, J.K., Osmotherly, L., Li, R., Liu, T., Zhang, Z., Bolund, L., Gane, Zheng, W., Dehal, P., Wang, J. & Durbin, R. (2006). TreeFam: a curated database of phylogenetic trees of animal gene families. *Nucleic Acids Research*, **34**, D572–D580. 112

Li, L., Stoeckert, C.J. & Roos, D.S. (2003). OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Research*, **13**, 2178–2189. 20

Li, Z., Zhang, S., Wang, Y., Zhang, X.S.S. & Chen, L. (2007). Alignment of molecular networks by integer quadratic programming. *Bioinformatics*, 1631–1639. 9

Lin, D. (1998). An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, 296–304, Morgan Kaufmann, San Francisco, CA. 76

Liu, Y., Liu, N. & Zhao, H. (2005). Inferring protein-protein interactions through high-throughput interaction data from diverse organisms. *Bioinformatics*, **21**, 3279–3285. 11, 74, 87, 129

Marcotte, E., Pellegrini, M., Ng, H., Rice, D., Yeates, T. & Eisenberg, D. (1999). Detecting protein function and protein-protein interactions from genome sequences. *Science*, **285**, 751–753. 10

Matthews, L.R., Vaglio, P., Reboul, J., Ge, H., Davis, B.P., Garrels, J., Vincent, S. & Vidal, M. (2001). Identification of potential interaction networks using sequence-based searches for conserved protein-protein interactions or "interologs". *Genome Research*, **11**, 2120–2126. 11

Medina, M. (2005). Genomes, phylogeny, and evolutionary systems biology. *Proc Natl Acad Sci U S A*, **102**, 6630–6635. 13

Mewes, H.W., Frishman, D., Mayer, K.F., Mnsterktter, M., Noubibou, O., Pagel, P., Rattei, T., Oesterheld, M., Ruepp, A. & Stmpflen, V. (2006). MIPS: analysis and annotation of proteins from whole genomes in 2005. *Nucleic Acids Research*, **34**, D169–D172. 63, 81, 130

Michaut, M., Kerrien, S., Montecchi-Palazzi, L., Chauvat, F., Cassier-Chauvat, C., Aude, J.C.C. & Legrain, P. (2008). InteroPorc: Automated inference of highly conserved protein interaction networks. *Bioinformatics*, **24**, 1625–1631. 76, 87

Mrowka, R., Patzak, A. & Herzel, H. (2001). Is there a bias in proteome research? *Genome Research*, **11**, 1971–1973. 5

Murphy, K. (1999). Pearl's algorithm for multiplexer nodes, http://www.cs.ubc.ca/~murphyk/Papers/pearlmux.ps.gz. 41

Narayanan, M. & Karp, R.M. (2007). Comparing protein interaction networks via a graph match-and-split algorithm. *Journal of Computational Biology*, **14**, 892–907. 9

Neapolitan, R.E. (2003). *Learning Bayesian Networks*. Prentice Hall. 36, 37, 38, 41

Needleman, S.B. & Wunsch, C.D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, **48**, 443–453. 17

Nei, M. (1996). Phylogenetic analysis in molecular evolutionary genetics. *Annual Review of Genetics*, **30**, 371–403. 22

NOTREDAME, C., HIGGINS, D.G. & HERINGA, J. (2000). T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology*, **302**, 205–217. 20

OHNO, S. (1970). *Evolution by gene duplication*. Springer Verlag. 32

PAGE, R. & CHARLESTON, M. (1997). From gene to organismal phylogeny: reconciled trees and the gene tree/species tree problem. 24, 50

PANCHENKO, A. & PRZYTYCKA, T., eds. (2008). *Protein-protein Interactions and Networks: Identification, Computer Analysis, and Prediction*, Springer. 6, 155, 156, 157

PASTOR-SATORRAS, R., SMITH, E. & SOL, R.V. (2003). Evolving protein interaction networks through gene duplication. *Journal of Theoretical Biology*, **222**, 199–210. 32, 53

PATIL, A. & NAKAMURA, H. (2005). Filtering high-throughput protein-protein interaction data using a combination of genomic features. *BMC Bioinformatics*, **6**, 100. 128

PAZOS, F. & VALENCIA, A. (2001). Similarity of phylogenetic trees as indicator of protein-protein interaction. *Protein Engineering*, **14**, 609–614. 11

PEARL, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann. 41, 57

PELLEGRINI, M., MARCOTTE, E., THOMPSON, M., EISENBERG, D. & YEATES, T. (1999). Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc Natl Acad Sci U S A*, **96**, 4285–4288. 10

PENROSE, M. (2003). *Geometric Random Graphs*. Oxford University Press. 34

PEVZNER, P.A. (2000). *Computational Molecular Biology*. The MIT Press. 16

Pinney, J.W., Amoutzias, G.D., Rattray, M. & Robertson, D.L. (2007). Reconstruction of ancestral protein interaction networks for the bZIP transcription factors. *Proc Natl Acad Sci U S A*, **104**, 20449–20453. 68

Prasad, K.T.S., Goel, R., Kandasamy, K., Keerthikumar, S., Kumar, S., Mathivanan, S., Telikicherla, D., Raju, R., Shafreen, B., Venugopal, A., Balakrishnan, L., Marimuthu, A., Banerjee, S., Somanathan, D.S., Sebastian, A., Rani, S., Ray, S., Kishore, H.C.J., Kanth, S., Ahmed, M., Kashyap, M.K., Mohmood, R., Ramachandra, Y.L., Krishna, V., Rahiman, A.B., Mohan, S., Ranganathan, P., Ramabadran, S., Chaerkady, R. & Pandey, A. (2008). Human Protein Reference Database – 2009 update. *Nucleic Acids Research*, **37**, D767–D772. 130

Przulj, N., Corneil, D.G. & Jurisica, I. (2004). Modeling interactome: scale-free or geometric? *Bioinformatics*, **20**, 3508–3515. 34

Pu, S., Wong, J., Turner, B., Cho, E. & Wodak, S.J. (2009). Up-to-date catalogues of yeast protein complexes. *Nucleic Acids Research*, 825–831. 81, 130

Reguly, T., Breitkreutz, A., Boucher, L., Breitkreutz, B.J., Hon, G., Myers, C., Parsons, A., Friesen, H., Oughtred, R., Tong, A., Stark, C., Ho, Y., Botstein, D., Andrews, B., Boone, C., Troyanskya, O., Ideker, T., Dolinski, K., Batada, N. & Tyers, M. (2006). Comprehensive curation and analysis of global interaction networks in Saccharomyces cerevisiae. *Journal of Biology*, **5**. 81, 130

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 448–453. 76

Rual, J.F., Venkatesan, K., Hao, T., Hirozane-Kishikawa, T., Dricot, A., Li, N., Berriz, G.F., Gibbons, F.D., Dreze, M., Ayivi-Guedehoussou, N., Klitgord, N., Simon, C., Boxem, M., Milstein, S., Rosenberg, J., Goldberg, D.S., Zhang, L.V., Wong, S.L., Franklin, G., Li, S., Albala, J.S., Lim, J., Fraughton, C., Llamosas, E., Cevik, S., Bex, C., Lamesch, P., Sikorski, R.S., Vandenhaute, J., Zoghbi, H.Y., Smolyar, A., Bosak, S., Sequerra, R., Doucette-Stamm, L., Cusick, M.E., Hill, D.E., Roth, F.P. & Vidal, M. (2005). Towards a proteome-scale map of the human protein-protein interaction network. *Nature*, **437**, 1173–1178. 2

Salwinski, L., Miller, C.S., Smith, A.J., Pettit, F.K., Bowie, J.U. & Eisenberg, D. (2004). The Database of Interacting Proteins: 2004 update. *Nucleic Acids Research*, **32**, D449–D451. 59, 74, 127

Sarnowski, T.J., Swiezewski, S., Pawlikowska, K., Kaczanowski, S. & Jerzmanowski, A. (2002). AtSWI3B, an Arabidopsis homolog of SWI3, a core subunit of yeast Swi/Snf chromatin remodeling complex, interacts with FCA, a regulator of flowering time. *Nucleic Acids Research*, **30**, 3412–3421. 77

Sarnowski, T.J., Ríos, G., Jásik, J., Swiezewski, S., Kaczanowski, S., Li, Y., Kwiatkowska, A., Pawlikowska, K., Kozbiał, M., Kozbiał, P., Koncz, C. & Jerzmanowski, A. (2005). SWI3 subunits of putative SWI/SNF chromatin-remodeling complexes play distinct roles during Arabidopsis development. *The Plant Cell*, **17**, 2454–2472. 77

Sauer, U., Heinemann, M. & Zamboni, N. (2007). Genetics: Getting closer to the whole picture. *Science*, **316**, 550–551. 2

Schlicker, A., Domingues, F., Rahnenführer, J. & Lengauer, T. (2006). A new measure for functional similarity of gene products based on Gene Ontology. *BMC Bioinformatics*, **7**, 302. 76, 78

Schwartz, A.S., Yu, J., Gardenour, K.R., Finley Jr, R.L. & Ideker, T. (2008). Cost-effective strategies for completing the interactome. *Nature Methods*, **6**, 55–61. 5

Shannon, P., Markiel, A., Ozier, O., Baliga, N.S., Wang, J.T., Ramage, D., Amin, N., Schwikowski, B. & Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, **13**, 2498–2504. 4, 121

SHARAN, R. & IDEKER, T. (2006). Modeling cellular machinery through biological network comparison. *Nature Biotechnology*, **24**, 427–433. 7, 10, 12, 13

SHARAN, R., IDEKER, T., KELLEY, B., SHAMIR, R. & KARP, R.M. (2005a). Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. *Journal of Computational Biology*, **12**, 835–846. 7, 58, 63, 64

SHARAN, R., SUTHRAM, S., KELLEY, R.M., KUHN, T., MCCUINE, S., UETZ, P., SITTLER, T., KARP, R.M. & IDEKER, T. (2005b). Conserved patterns of protein interaction in multiple species. *Proc Natl Acad Sci U S A*, **102**, 1974–1979. 7, 12, 64, 66, 86

SHIM, S., MERRILL, S.A. & HANSON, P.I. (2008). Novel interactions of ESCRT-III with LIP5 and VPS4 and their implications for ESCRT-III disassembly. *Molecular Biology of the Cell*, **19**, 2661–2672. 77, 94

SHOEMAKER, B. & PANCHENKO, A. (2007a). Deciphering protein-protein interactions. Part II. Computational methods to predict protein and domain interaction partners. *PLoS Computational Biology*, **3**, e43. 10

SHOEMAKER, B.A. & PANCHENKO, A.R. (2007b). Deciphering protein-protein interactions. Part I. Experimental techniques and databases. *PLoS Computational Biology*, **3**, e42. 3, 4

SINGH, R., XU, J. & BERGER, B. (2007). Pairwise global alignment of protein interaction networks by matching neighborhood topology. In T.P. Speed & H. Huang, eds., *RECOMB*, vol. 4453 of *Lecture Notes in Computer Science*, 16–31, Springer. 9

SMITH, T.F. & WATERMAN, M.S. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, **147**, 195–197. 18

SOLE, R.V., PASTOR-SATORRAS, R., SMITH, E. & KEPLER, T.B. (2002). A model of large-scale proteome evolution. *Advances in Complex Systems*, **5**, 43–54. 32, 35, 53, 60

Stelzl, U., Worm, U., Lalowski, M., Haenig, C., Brembeck, F.H., Goehler, H., Stroedicke, M., Zenkner, M., Schoenherr, A., Koeppen, S., Timm, J., Mintzlaff, S., Abraham, C., Bock, N., Kietzmann, S., Goedde, A., Toksöz, E., Droege, A., Krobitsch, S., Korn, B., Birchmeier, W., Lehrach, H. & Wanker, E.E. (2005). A human protein-protein interaction network: a resource for annotating the proteome. *Cell*, **122**, 957–968. 2

Studier, J. & Keppler, K. (1988). A note on the neighbor-joining algorithm of Saitou and Nei. *Molecular Biology and Evolution*, **5**, 729–731. 23

Stumpf, M.P., Thorne, T., de Silva, E., Stewart, R., An, H.J., Lappe, M. & Wiuf, C. (2008). Estimating the size of the human interactome. *Proc Natl Acad Sci U S A*, **105**, 6959–6964. 5, 84, 129

Tatusov, R.L., Fedorova, N.D., Jackson, J.D., Jacobs, A.R., Kiryutin, B., Koonin, E.V., Krylov, D.M., Mazumder, R., Mekhedov, S.L., Nikolskaya, A.N., Rao, B.S., Smirnov, S., Sverdlov, A.V., Vasudevan, S., Wolf, Y.I., Yin, J.J. & Natale, D.A. (2003). The COG database: an updated version includes eukaryotes. *BMC Bioinformatics*, **4**, 41. 59

Tiuryn, J. (2006). Introduction to computational biology (lecture notes in Polish), University of Warsaw. 16

Uetz, P., Giot, L., Cagney, G., Mansfield, T., Judson, R., Knight, J., Lockshon, D., Narayan, V., Srinivasan, M., Pochart, P., Qureshi-Emili, A., Li, Y., Godwin, B., Conover, D., Kalbfleisch, T., Vijayadamodar, G., Yang, M., Johnston, M., Fields, S. & Rothberg, J. (2000). A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, **403**, 623–627. 2, 5, 74

Uetz, P., Titz, B. & Cagney, G. (2008). Experimental methods for protein interaction identification and characterization. In Panchenko & Przytycka (2008), 1–32. 4

Valencia, A. & Pazos, F. (2002). Computational methods for the prediction of protein interactions. *Current Opinion in Structural Biology*, **12**, 368–373. 10

Valencia, A. & Pazos, F. (2008). Computational methods to predict protein interaction partners. In Panchenko & Przytycka (2008), 67–81. 10

Vazquez, A., Flammini, A., Maritan, A. & Vespignani, A. (2003). Modeling of protein interaction networks. *Complexus*, **1**, 38–44. 34

Venkatesan, K., Rual, J.F., Vazquez, A., Stelzl, U., Lemmens, I., Hirozane-Kishikawa, T., Hao, T., Zenkner, M., Xin, X., Goh, K.I., Yildirim, M.A., Simonis, N., Heinzmann, K., Gebreab, F., Sahalie, J.M., Cevik, S., Simon, C., de Smet, A.S., Dann, E., Smolyar, A., Vinayagam, A., Yu, H., Szeto, D., Borick, H., Dricot, A., Klitgord, N., Murray, R.R., Lin, C., Lalowski, M., Timm, J., Rau, K., Boone, C., Braun, P., Cusick, M.E., Roth, F.P., Hill, D.E., Tavernier, J., Wanker, E.E., Barabási, A.L. & Vidal, M. (2009). An empirical framework for binary interactome mapping. *Nature Methods*, **6**, 83–90. 5

Venter, J.C., Adams, M.D., Myers, E.W., Li, P.W., Mural, R.J., Sutton, G.G., Smith, H.O., Yandell, M., Evans, C.A., Holt, R.A., Gocayne, J.D., Amanatides, P., Ballew, R.M., Huson, D.H., Wortman, J.R., Zhang, Q., Kodira, C.D., Zheng, X.H., Chen, L., Skupski, M., Subramanian, G., Thomas, P.D., Zhang, J., Miklos, G.G.L., Nelson, C., Broder, S., Clark, A.G., Nadeau, J., Mckusick, V.A., Zinder, N., Levine, A.J., Roberts, R.J., Simon, M. & Slayman, C. (2001). The sequence of the human genome. *Science*, **291**, 1304–1351. 1

von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S., Fields, S. & Bork, P. (2002). Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, **417**, 399–403. 5

Wagner, A. (2001). The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes. *Mol Biol Evol*, **18**, 1283–1292. 55, 60

WANG, L. & JIANG, T. (1994). On the complexity of multiple sequence alignment. *Journal of Computational Biology*, **1**, 337–348. 19

WATTS, D.J. & STROGATZ, S.H. (1998). Collective dynamics of 'small-world' networks. *Nature*, **393**, 440–442. 28

YOOK, S.H., OLTVAI, Z.N. & BARABÁSI, A.L. (2004). Functional and topological characterization of protein interaction networks. *Proteomics*, **4**, 928–942. 28

YOSEF, N., RUPPIN, E. & SHARAN, R. (2008). Cross-species analysis of protein-protein interactions networks. In Panchenko & Przytycka (2008), 163–185. 7

YU, H., BRAUN, P., YILDIRIM, M.A., LEMMENS, I., VENKATESAN, K., SAHALIE, J., HIROZANE-KISHIKAWA, T., GEBREAB, F., LI, N., SIMONIS, N., HAO, T., RUAL, J.F., DRICOT, A., VAZQUEZ, A., MURRAY, R.R., SIMON, C., TARDIVO, L., TAM, S., SVRZIKAPA, N., FAN, C., DE SMET, A.S., MOTYL, A., HUDSON, M.E., PARK, J., XIN, X., CUSICK, M.E., MOORE, T., BOONE, C., SNYDER, M., ROTH, F.P., BARABASI, A.L., TAVERNIER, J., HILL, D.E. & VIDAL, M. (2008). High-quality binary protein interaction map of the yeast interactome network. *Science*, **3**, 104–110. 5, 81, 86, 130