Algorithms for the Analysis of Protein Interaction Networks

by

Rohit Singh

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2012

Author .....
Department of Electrical Engineering and Computer Science
September 12, 2011

Certified by .....
Bonnie Berger
Professor
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Students

Algorithms for the Analysis of Protein Interaction Networks

by

Rohit Singh

Submitted to the Department of Electrical Engineering and Computer Science
on September 12, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

In the decade since the human genome project, a major research trend in biology has been towards understanding the cell as a system. This interest has stemmed partly from a deeper appreciation of how important it is to understand the emergent properties of cellular systems (e.g., they seem to be the key to understanding diseases like cancer). It has also been enabled by new high-throughput techniques that have allowed us to collect new types of data at the whole-genome scale.

We focus on one sub-domain of systems biology: the understanding of protein interactions. Such understanding is valuable: interactions between proteins are fundamental to many cellular processes. Over the last decade, high-throughput experimental techniques have allowed us to collect a large amount of protein-protein interaction (PPI) data for many species. A popular abstraction for representing this data is the protein interaction network: each node of the network represents a protein and an edge between two nodes represents a physical interaction between the two corresponding proteins. This abstraction has proven to be a powerful tool for understanding the systems aspects of protein interaction.

We present some algorithms for the augmentation, cleanup and analysis of such protein interaction networks:

1. In many species, the coverage of known PPI data remains partial. Given two protein sequences, we describe an algorithm to predict if two proteins physically interact, using logistic regression and insights from structural biology. We also describe how our predictions may be further improved by combining with functional-genomic data.

2. We study systematic false positives in a popular experimental protocol, the Yeast 2-Hybrid method. Here, some "promiscuous" proteins may lead to many false positives. We describe a Bayesian approach to modeling and adjusting for this error.

3. Comparative analysis of PPI networks across species can provide valuable insights. We describe IsoRank, an algorithm for global network alignment of multiple PPI networks. The algorithm first constructs an eigenvalue problem that encapsulates the network and sequence similarity constraints. The solution of the problem describes a $k-$partite graph that is further processed to find the alignment.

4. For a given signaling network, we describe an algorithm that combines RNA-interference data with PPI data to produce hypotheses about the structure of the signaling network. Our algorithm con-

structs a multi-commodity flow problem that expresses the constraints described by the data and finds a sparse solution to it.

Thesis Supervisor: Bonnie Berger
Title: Professor

To my parents, Sarswati and Mahendra Pratap Singh

Acknowledgments

This thesis would not have been possible without the support and encouragement of many people. I mention some in particular here, and apologize to the rest for the lack of space for acknowledging them.

I thank my advisor, Bonnie Berger, for her unswerving belief in me and constant encouragement, guidance and mentoring. Over the years, Bonnie has mentored a truly remarkable group of computational biology researchers. Looking back and extrapolating from my own experience, I can appreciate how effective Bonnie is at bringing out the best in her students. She would leave me alone for weeks when I was coding furiously, but be available for a 3 A.M. editing discussion on the night of a conference deadline. She would let me pick out the problems I wanted to work on while teaching me how to identify what made for a promising topic. She also taught me how to augment, amplify and advocate for my work. Much of the work presented in this thesis may well have ended as small, isolated projects if Bonnie had not nudged— and when needed, pushed— me to round out the work, explore the connections between them, and make it available as a community resource. Along the way, Bonnie also taught me how to build good collaborative research relationships. Finally, I am grateful for her flexibility when I went on non-resident status and was often available only over phone or email.

I also thank my other thesis committee members, David Gifford and Srini Devadas. I would like to thank David Gifford for his guidance and mentorship in my first year at MIT and for his wise and friendly advice in the years since. I am also thankful to him for one of the most fun classes I have ever taken and for helping two newbie grad students (Nathan Palmer and I) expand a class project into a full-blown research paper. I am also deeply appreciative of Srini Devadas for his advice and feedback and for making the time to see me whenever I requested.

There are many other faculty members who have been a guiding force over the years. At IIT Kanpur, Amitabha Mukerjee introduced me to the joys of doing research. At Stanford, my Masters advisor Jean-Claude Latombe provided me with the first introduction to computational biology and gave me valuable lessons about both research and life. I am grateful to Tommi Jaakkola (MIT) for letting me TA his machine learning class and learn a huge amount in the process. A few times when I was stuck on some algorithmic issues, I met with Tom Leighton (MIT). Tom's clarity of thought has always amazed me and is something I have tried hard to mimic. Finally, I want to thank Norbert Perrimon (Harvard Medical School) for letting me into his group meetings, teaching me biology, listening patiently to my naive questions about why things were done the way they were, and yet being able to see (and show me) how the use of computational

methods could change that.

Members of the Berger Group have been a great source of help, fun and support. Patrice Macaluso has been extremely supportive, patiently dealing with all my various pleas for administrative help. Nathan Palmer is a collaborator and a valued friend, commiserating with me over research setbacks and celebrating life's milestones. Jinbo Xu has been a collaborator, a friend and a mentor. His efficiency and work-ethic are an inspiration. Patrick Schmid and Allen Bryan have provided for many fun discussions over the years. Some of my happiest memories of MIT involve random discussions with Allen, in which people from the adjoining room (Nathan, Patrick, Michael) would join in occasionally. It has been a joy collaborating with Michael Baym, Raghu Hosur, Irene Kaplow, Daniel Park, David Sontag, Beckett Sterner and George Tucker. It has also been fun working and hanging out with Michael Schnall-Levin, Shannon Wieland, Oaz Nir, Luke Hutchison and Vinay Pulim.

In the Perrimon Lab at Harvard, it has been a pleasure talking about research with Adam Friedman and Chris Bakal. I prize the friendship I have developed with Srikanth Kandula in my years at MIT. In the early years of the PhD, my friendship with Srikanth and Sachin Katti helped keep me sane. From 2007 to 2011, I have been on non-resident status, working in New York. I thank my manager, Parag Tole, and my colleagues Rahul Sahni and Sasha Oblak for their patience and accommodation during the times when I had to devote my attention to research work and writing.

My brother, Rajeev, has been a guiding light through my life. His achievements have inspired me; his advice has helped me figure out what I want to do with my life; and his unconditional support has helped me recover from life's setbacks, little and large. My sister Preeti, my sister-in-law Anamika, my brother-in-law Shivendra and my cousin Vivek have had unwavering belief in me, helping me get over the periods when I felt down. My daughter Ira was born during my PhD. I could not have imagined a more joyful distraction from my PhD. This thesis would not have been possible without my wife Sanskriti's support and faith in me during the many years it has taken me to finish it. After the umpteenth time I canceled on any weekend plans so I could work on the thesis and then spent the weekend procrastinating, a less saintly wife would have lost her patience. Finally, my parents, Sarswati and Mahendra Pratap Singh, have been a pillar of support all through my life. Their belief that I could and should do good research kept me motivated through the years of my PhD. Without their love, sacrifices and encouragement, I would have neither gotten to MIT nor finished this thesis. This thesis is dedicated to them.

Previously Published Work

Some of the material in this thesis has been published in a journal or presented at a conference previously:

1. Material from Chap. 3 was presented at the Pacific Symposium of Biocomputation (PSB) in 2006 and published in Nucleic Acids Research (Vol 38) in 2010.

2. Material from Chap. 4 was presented at the Pacific Symposium of Biocomputation (PSB), 2007.

3. Material from Chap. 5 was presented at the Research in Computational Molecular Biology (RECOMB) in 2007, at the Pacific Symposium of Biocomputation (PSB) in 2008, and International Conference on Intelligent Systems for Molecular Biology (ISMB) in 2009. It was also published in Proceedings of National Academy of Sciences (Vol 105) in 2008 and in Bioinformatics (Vol 25) in 2009.

4. Material from Chap. 6 was presented at the International Conference on Intelligent Systems for Molecular Biology (ISMB) in 2007.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the decade since the human genome was first sequenced, research in biology has exploded, aided by improvements in both experimental techniques and computational analyses. In some ways, the data from these studies has muddied the waters. For example, linear cascades were a popular way of modeling signal transduction; now it seems that signals follow much more complex paths inside the cell [50]. It turns out that the importance of RNA (e.g., microRNAs and siRNAs) in the regulatory process had been severely underestimated [7]. Similarly, the regulation of gene expression seems to be significantly more multi-modal than earlier thought, involving chromatin remodeling, an array of repressors and activators, post-transcriptional regulation of mRNA, translational regulation at the ribosome and so on [24, 45]. The story clearly seems a lot more complicated than we imagined a decade ago.

Viewed from another perspective, though, all these studies point us towards the same core insight: the cell is a system. While it has long been known that the components of a cell act together as part of a system, the experiments performed over the last decade have helped us appreciate how truly deep the interconnections are. The cell is an amazingly sophisticated system, robust in many ways and yet, surprisingly fragile in others. It consists of various inter-connected sub-systems, each rather complicated in its own right [57]. For example, the signal transduction machinery in a cell influences and is influenced by the transcriptional regulatory framework, which in turn may be influenced by a variety of microRNAs. Each of these sub-systems displays many of the control elements seen in man-made electrical/mechanical systems: feedback control, signal integration, signal amplification etc.

Understanding the cell as a system has become one of the most active areas of research within biology. Such an understanding will provide significant practical benefits. For example, many diseases have causes

that are systems-related. Certain kinds of cancer happen when the signaling/regulatory mechanisms of the cell that control cell growth, reproduction and death develop a malfunction, leading to uncontrolled cell growth [85, 27, 80]. The specific malfunction varies across different kinds of cancers; indeed, this is a key reason why a single cure for cancer has been so elusive. Even for a disease like diabetes, where the basic cause is well-known, there are systems-related subtleties that remain to be explained. For example, studies suggest that only a subset of individuals who are insulin-resistant actually go on to develop adult-onset diabetes. The difference seems to be that the insulin-producing pancreatic β-cells of susceptible individuals fail to proliferate and are thus unable to produce enough insulin to compensate for the insulin resistance [8]. It is not clear why this happens.

Systems biology may also help deepen our understanding of evolutionary biology. One of the key goals there is to understand how a gene (or a family of genes) has evolved across species. A natural extension of this problem is understanding how cellular systems evolve across species. To see how such insights may be valuable, consider the following open problem.

One of the surprising discoveries of the human genome project was the relatively low number of human genes. Before the project, the total gene-count in humans was expected to be about 100,000 [30]. However, recent analyses of the human genome estimate this count to be much lower— in the 20,000-25,000 range [46]. This is not much more than the gene-count of the fruit fly and about the same as that of the worm. So how come humans are the ones doing experiments on worms and not vice-versa? In other words, where does organismal complexity arise from? A partial explanation seems to be that the proteins, genes, and RNA in human cells are part of a more complex system than the corresponding ones in fruit fly or worm [18, 75]. A comparative analysis of the cellular systems in various species could thus provide valuable insights.

1.1 Understanding Protein Interaction

In this thesis, we focus on a specific domain within systems biology: the elucidation and analysis of protein-protein interactions. Proteins are the workhorses of the cell. The genetic information encoded in DNA (or RNA, in some cases) is transcribed and translated to produce proteins which then carry out the vast majority of tasks within the cell: metabolism, signal transduction, vesicle transport etc. However, proteins do not act in isolation. They perform their function in the context of other proteins, by influencing and interacting with them. Understanding protein interactions is thus crucial to understanding protein

function. Historically, such interactions have been studied from a *worm's eye* view, with the goal being to get a deep mechanistic understanding of how a particular pair of proteins interacts. The extensive work on protein docking and ligand-protein binding has followed this approach.

In contrast, systems biologists take a *bird's eye* view approach towards understanding protein interactions. They emphasize the importance of understanding the general pattern of protein interactions across all the proteins, rather than focusing on just a few interactions [21]. This approach has been enabled by the advent of high-throughput methods for discovering protein interactions which have led to an exponential growth in the sizes of PPI datasets. Data from these experiments has been accumulating at an extremely rapid pace over the last decade. For example, the data for humans now covers about 50,000 PPIs involving 10,000 proteins. While such coverage is by no means comprehensive, analysis of currently-available data has already led significant insights about the cellular system.

A useful model for organizing this data is the protein-protein interaction network: a graph where each node corresponds to a protein and an edge between two nodes indicates a direct physical interaction between the corresponding proteins. Thus, for the human genome, the graph will have about 10,000 nodes and 50,000 edges. Analysis of these PPI networks has already yielded some valuable biological insights. For example, their topological analysis suggested that the node-degree distribution in these networks follows a power-law distribution, rather than a uniform distribution. This, in turn, immediately suggests that certain nodes— those with high degrees (*hubs*)— play a disproportionately important role in the cell and are crucial to the connectivity in the PPI graph. In experiments, the removal (*knocking-out*) of a hub protein led to significantly deleterious effects. In contrast, removal of the low-degree proteins (which constitute the vast majority) had a much weaker impact [52]. This ties in nicely with a long-observed biological phenomenon: while cellular systems are typically robust to many kinds of (fairly drastic!) environmental changes, they are surprisingly fragile with respect to other — seemingly minor — changes. The PPI network's topology suggests a possible reason. Random attacks on the cellular system will most likely disrupt one of the non-hub proteins (these are the most numerous) and have a relatively low impact; on the other hand, directed attacks which hit a hub protein can have a far greater impact.

A combination of PPI networks with other kinds of functional genomic data has proven to be especially informative. For example, Han *et al.* [42] have combined PPI data with gene expression to classify hub proteins into two classes: date hubs and party hubs. The former are hub proteins with relatively low expression similarity with their neighbors while the latter have high expression similarity with their neighbors. This immediately suggests that the party hub proteins form the scaffolding of a multi-protein

complex while the date hub proteins are signal carriers, transmitting information down a signaling cascade.

1.2 Contributions

In this thesis, we propose to investigate some of the many challenges in the computational analysis of biological networks. Broadly speaking, we are interested in answering the following questions about PPI data:

1. Is this data completely reliable? If not, can we improve its quality?
2. What biological insights can one derive from this data?
3. Can this data be combined with other biological data for further insights?

These questions can be formulated as problems that are both computationally interesting as well biologically valuable. In this work, we focus on PPI networks, but some of the methods proposed here extend to other kinds of biological networks (e.g. protein-DNA networks) as well.

The first such challenge is to address data quality issues in experimental PPI data. Experimental PPI data suffers from both false negatives and false positives [90]. The false negatives arise from (1) lack of coverage, i.e., enough experiments required to test all possible interactions have not been performed and (2) shortcomings in experiment design due to which certain *in-vivo* interactions are not observed *in-vitro*. On the other hand, PPI data also has many false positives. These arise from limitations of the experimental setup due to which a pair of proteins is reported to interact even though they actually might not interact *in-vivo*. To address these issues we propose two algorithms. The first predicts PPIs computationally to improve the coverage of current PPI data. The key contribution of the algorithm is to use structure-based methods to predict interaction between proteins, given only their sequence information. Towards identifying false positives in PPI data, we describe a probabilistic relational model to identify false positives in data from Two-Hybrid (2H) experiments, one of the two commonly used high-throughput methods to infer PPI. Our method models both random as well as systematic errors in 2H data and was the first method to do so.

We next focus on deriving concrete biological insights from PPI data. Specifically, we use it to better estimate sets of genes that perform the same function in various species. Until recently, such comparative genomic analyses have been performed using only sequence data. However, PPI data provides a functional

perspective, and its use in such comparative analyses may provide new insights [81]. We investigate the following problem: given two or more PPI networks (corresponding to different species), find the best overall alignment of the networks, taking into account both the network topologies as well as sequence similarities between the individual proteins of the networks. This network alignment problem is analogous to the global sequence alignment problem— we are interested in the best overall match between the two inputs. We propose an algorithm for this problem that relies on the following intuition: a node X in network N_1 is a good match for X' in network N_2 if and only if the neighbors of X are good matches for the neighbors of X'. To formalize this intuition, we construct an eigenvalue problem whose results are then used to construct a bipartite graph. Solving a max-weight matching problem on this graph produces the desired mapping. We use our method to predict functional orthologs, i.e., pairs of proteins (in two species) that perform the same function. It can also be extended to perform multiple network alignment. Using our multiple network alignment algorithm, we can produce orthology mappings that, by some measures, are more biologically accurate than current orthology lists.

The final problem we consider is the generation of high-confidence hypotheses about the topology of a signaling network by integrating PPI data with RNA interference (RNAi) data. In particular, we make use of pathway-specific RNAi experiments. In such experiments, the end-effector gene of a given signaling pathway (e.g. *Erk* in the MAP Kinase pathway) is chosen. Then, using RNAi, every other gene in the genome is knocked down one-by-one, and the effect on the reporter gene's activity is measured. Such experiments provide a list of genes (*hits*) that influence the level of the reporter gene and, for each such hit, a measure of the relative strength of its influence on the reporter gene [35]. Our algorithm is driven by parsimony considerations: it searches for the simplest directed graph that is consistent with the observed PPI and RNAi data and also with the known biology of the given pathway. It generates a directed, sparse (tree-like) graph whose nodes correspond to RNAi hits and whose edges may be interpreted as high-confidence hypotheses about the signaling network's structure. We begin by constructing a constructing an integer linear program (ILP), borrowing ideas from the multicommodity network flow literature to represent the biological constraints. We relax the ILP to a linear program and solve it to produce the final output. Our method, though based on very simple constraints, suggests surprisingly plausible hypotheses. For example, we specified to it only a truncated version of the known core cascade for the MAPK pathway. Our algorithm not only recovered the remaining components of the core cascade but also suggested connections between these components that are consistent with our biological understanding of the MAPK cascade.

This thesis is organized as follows. The next chapter provides a very brief primer of, for computer scientists, the biological concepts and datasets used. The subsequent chapters focus on the creation (Chapter 3), cleanup (Chapter 4) and usage (Chapter 5 & 6) of PPI data. Chapter 3 aims to augment experimental PPI data by using protein structure to predict PPI data. The next chapter describes a systematic bias in the Two-Hybrid protocol for elucidating PPIs and a machine learning approach for mitigating it. In Chapter 5, we describe the first global alignment of PPI networks across multiple species, with direct implications for better prediction of gene correspondences across species. The next chapter proposes a novel approach to combining PPI data with RNA interference data, so as to better understand cell signaling.

Chapter 2

Background

In this chapter, we provide a brief background on the biological concepts relevant to this thesis. We also point out the publicly accessible biological databases that we used in the course of much of the research presented in this thesis.

2.1 The Key Players: DNA, RNA and Proteins

The most fundamental relationship in molecular biology is the one between DNA, RNA, and proteins. Each of these are polymeric chains formed by concatenation of simple molecules. Both DNA (Deoxyribonucleic acid) and RNA (Ribonucleic acid) are polymers of nucleotides. They differ in the composition of sugar molecules in their respective backbones and the set of nucleotides used in each. Proteins are polymers of amino acids. The variety of building blocks in each of these macromolecules is rather limited: DNA contains only 4 kinds of nucleotides; so does RNA. In most cases, proteins contain 20 (or fewer) kinds of amino acids. Instead, the complexity of these macromolecules arises from the number and ordering of building blocks in each. In fact, representing these macromolecules as just an abstract sequence of building blocks, without regard to the specific biochemistry, has proven quite valuable. In such an abstraction, these molecules are represented as strings of letters, with the alphabet sizes of 4, 4, and 20 for DNA, RNA and proteins respectively. Many biological problems can then be posed as string-based computational problems. For example, finding the human equivalent of a particular chimpanzee gene essentially reduces to the problem of finding, from a set of strings, the one most similar to a given string pattern.

The central dogma of molecular biology states that information in a cell flows from DNA to RNA and

then to proteins. The genetic information in a cell is typically encoded using DNA[1]. A *gene* is a sequence of DNA that contains the information needed to construct a single protein. This information is decoded and re-encoded into RNA, a process known as *transcription*. These messenger RNA (mRNA) molecules are produced inside the nucleus and are exported to the cytoplasm where protein synthesis takes place. In a process known as *translation*, the information in a mRNA molecule is read to produce a sequence of amino acids, which make up the protein. Each mRNA *codon*, a sequence of 3 nucleotides, corresponds to 1 amino acid in the protein.[2]

Later in this chapter, we will revisit the transcription and translation processes to describe certain aspects particularly relevant to this thesis (control of gene expression and control of mRNAs by RNA interference). We first discuss protein structure, function and interactions.

2.2 Proteins: From Sequence to Structure

Proteins are the workhorses of the cell. They comprise much of the structural scaffolding of the cell and play key roles in almost all intra-cellular activity: transport, signal transduction, metabolism, DNA replication etc. Even the processes for creating and recycling proteins rely on certain specialized proteins. In performing its function, a protein's structure is crucially important. One of the fascinating mysteries of the transcription and translation processes is that the sequential (1-dimensional) information from DNA, when re-encoded into a protein, leads to a specific 3-dimensional structure uniquely determined by the sequence. One of the open problems in computational biology — and a very active area of research — is understanding how this happens, i.e., how to predict a protein's structure given its sequence. In Chapter 3, we leverage insights from this research to predict if a pair of proteins will interact. Below, we briefly review some of those insights.

There have been two broad sets of approaches to protein structure prediction. One set of methods starts from first principles and aims to find the arrangement of atoms that will minimize the total free energy of the molecule. In these *ab-initio* methods, an energy function is constructed over the space of all possible

[1]Like most generalizations about biology, there are exceptions to this statement. Retroviruses (e.g., the AIDS-causing HIV virus) use RNA to encode their genetic information.

[2]Interestingly, the length of an mRNA codon (3 nucleotides per amino acid) is the smallest possible value which still ensures that (1) any protein's sequence can be encoded as RNA, and (2) any encoded RNA sequence represents a unique protein. A shorter coding scheme, e.g. 2 nucleotides per amino acid, could only represent $4^2 = 16$ unique amino acids. With a 3-letter codon, there are $4^3 = 64$ combinations, and all 20 amino acids can be covered, with a few left over to identify starting and stopping points.

conformations (structures) of the given protein; we then search this space for the conformation with the lowest energy. The challenge here is two-fold: even for very simplistic energy functions, the search problem is computationally difficult. Berger *et al.* [10] proved this for a simple HP-lattice model. Luckily, the search problem lends itself to parallelization and can be attacked using grid-computing methods. However, finding the optimal energy function (whose global minimum actually corresponds to the actual 3D structure) remains an open problem.

A second set of approaches aims to make use of experimental data available for proteins whose structures have been determined using crystallographic methods. If two proteins A and B are similar in sequence, their structures are likely to be similar as well. Assuming A's structure is known, such *homology modeling* approaches set the starting conformation of B to be A's known conformation and are predicated on the hypothesis that the B's globally optimum conformation is close enough to the starting conformation that it can be found by a local search around the latter. The challenges here are two-fold: finding a suitable base structure from which one can start and an optimization technique that can find the final structure given this base structure.

In recent years, attempts have been made to blend the homology modeling and *ab-initio* approaches. Baker *et al.* [14, 82, 15] has described an approach that uses existing structure to generate a library of small fragments of known structure, each such fragment being a few amino-acids long. The energy function for the global search problem operates on these fragments. In this thesis, we use some of the structure prediction work to evaluate pairs of proteins, aiming to evaluate if two protein structures are likely to form a joint structure complex.

2.3 Gene Transcription

Genetic information in the cell is typically encoded in DNA. This information is transcribed into RNA and then translated into proteins. To stop further production of a protein, either the gene for that protein can be turned off or the mRNA, once produced, be deactivated. There exist sophisticated cellular machinery for both these tasks, as well as for the opposite task of enhancing a protein's quantity. The study of the mechanisms (i.e., the transcriptional regulatory subsystem) by which all this happens is an extremely active of research. Here, we only mark out an aspect of this research that relates to this thesis.

Gene Expression Experiments: Over the last 15 years or so, one of the more powerful additions to a biologist's toolkit has been the gene expression experiment: the ability to simultaneously measure the

transcriptional activity level of many — or even all — genes in a given sample. Typically, the level of mRNA corresponding to each gene is measured. Given a tissue sample, the mRNA is first purified and then used to create complementary-DNA (cDNA) by reverse-transcription. One can then use specialized "gene-chips" that contain a well for each gene, with each well containing short DNA matching sequences ("probes") that can bind to the cDNA of a specific gene [79, 60]. While the measurement process remains significantly error-prone, the data from such experiments has provided valuable information. For example, such experiments can be used to identify *bio-markers* for some diseases, i.e., a set of genes whose abnormal activity level is an indicator that the cell is diseased.

By performing gene expression experiments on cells exposed to different conditions or at different points in their life-cycle, one can create *expression profiles* for each gene that summarize how its activity varies across conditions. Genes that are over-expressed (or under-expressed) under similar conditions will then have similar expression profile. It has been observed that such genes often correspond to proteins that interact with each other.

2.4 RNA Interference

To understand a gene's (and its protein's) role in the cellular system, perturbations experiments can be a powerful tool. In such experiments, the gene can either be *knocked-out* (e.g. removed from the genome entirely), or *knocked-down* (i.e., it's activity reduced). Knock-out experiments can be cumbersome to perform and can sometimes be too drastic a perturbation: a cell might not be viable if a particular gene is completely removed from its genome. However, for a long time, they were the only tool general enough to be used for genome-wide scans.

The advent of RNA interference (RNAi) experiments has significantly enhanced biologists' ability to perform genome-wide knock-down perturbation studies[3] This is done by utilizing the cellular machinery that uses small interfering RNA (siRNA) to regulate post-transcriptional gene activity. A double-stranded RNA (dsRNA) segment is introduced into the cell. It is then cleaved by Dicer, an enzyme, into its two constituent strands (the passenger and guide strands). The guide strand is taken up by the RNA-Induced Silencing Complex (RISC) which then uses it to recognize the complementary mRNA fragments inside the cell. The latter are subsequently broken down by the RNAase enzyme, thus neutralizing the corresponding

[3]Such has been the impact of RNAi in biology that within 8 years of discovering it [32], Andrew Fire and Craig Mello were awarded the Nobel Prize [98]. This was certainly one of the quicker Nobel Prizes.

gene's activity [97, 63]. Thus, by appropriately designing the dsRNA fragment, one can knock-down any desired gene. Libraries of such dsRNA fragments can be made to enable high-throughput genome-wide perturbation studies. We also note here that RNAi is simply the mechanism that enables perturbation; one still needs an assay to measure the appropriate cellular activity. Such an assay can be measure the activity of a single reporter gene (Friedman and Perrimon [37]), or be a microarray experiment measuring gene expression levels of multiple genes or even a measurement of cellular morphology (Nir *et al.* [69]).

2.5 Protein-Protein Interactions

The first proteins discovered were enzymes, found because of their ability to mediate and catalyze reactions between biomolecules. Thus, it has long been appreciated that a protein's function often involves interacting with other biomolecules. In this thesis, we focus specifically on protein-protein interactions. Until the early 2000's, the systematic study of protein-protein interactions had been difficult, mostly because of a lack of data. Some protein-protein interaction data *was* collected by co-crystallizing protein pairs and complexes and discerning their structure via X-ray crystallography. This has provided valuable insights into the structural mechanics of protein interaction (we use some of these insights in Chapter 3). However, crystallizing a single protein is difficult enough. It is even more difficult to co-crystallize protein complexes. Consequently, not very many protein complexes were found in this way.[4]

The systematic study of protein-protein interactions has had to wait for the advent of experimental techniques that allow for the discovery of new PPIs in a high-throughput approach. To get an idea of why these methods are thought of as "high-throughput", it is useful to think of their algorithmic complexity. What makes PPI detection hard is that it is a set of $O(n^2)$ problem instances, where n is the number of proteins in the genome of interest. Earlier methods approached each of these as a separate, slow task. In contrast, both the methods below exploit the problem's structure to first perform per-protein pre-processing steps. This may be relatively slow but there are only $O(n)$ such tasks. The pre-processing then enables $O(n^2)$ PPI elucidation tasks to be performed much more quickly than before. As a result of these approaches, the corpus of known PPIs has increased tremendously over the last decade (Fig 2-1). Here, we briefly describe the two commonly-used high-throughput approaches for discovering PPIs. A brief understanding of how they work will help clarify how some of the work described in this thesis helps

[4]One related area that did receive a lot of attention was the subject of protein/small-molecule binding. This is particularly relevant from a drug-discovery perspective, as drug-makers often aim to understand how a particular target protein may be bound to (and neutralized by) small molecules.

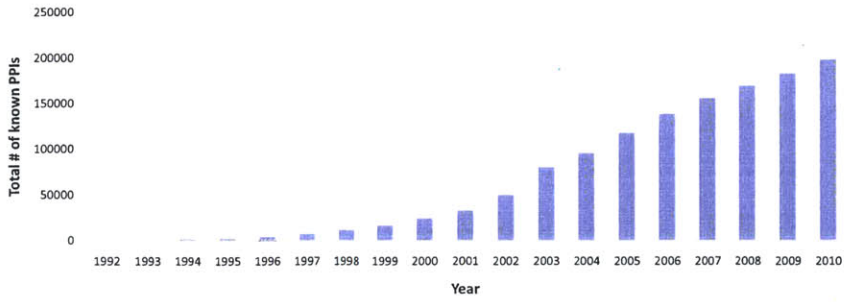


Figure 2-1: **The growth of known PPI data over the last decade.** With the advent of high-throughput techniques, the corpus of known PPIs (as measured from publication date of entries in BIOGRID[83]) grew exponentially over the early 2000's. In the later part of the decade, the growth has slowed down somewhat.

complement them or address their shortcomings.

2.5.1 2-Hybrid Techniques

The key idea here is to design an assay where a reporter gene will be turned on (with easily observable effects) if two protein A and B interact. The main challenge here is to design the assay in a way that is scalable to a genome-wide study. Towards this, Fields and Song [31] designed an ingenious scheme. They began by selecting a gene in yeast whose activation has two properties: 1) it can be easily observed (e.g., it results in galactose uptake by the cell), 2) its activation requires a transcription-factor protein with two structural domains: a DNA-binding domain (BD and an activation domain (AD). Given the two query proteins A and B, we then construct cloning vectors[5], one with the BD part fused to A and another with the AD part fused to B. These vectors are then introduced into the yeast cell and expressed (i.e. their protein is produced). Now, if A and B physically interact, the BD and AD domains will be in proximity and the reporter gene will be turned on. The reason this approach scales well to genome-scale studies [47] is that one can generate "libraries" of cloning vectors, each containing the gene sequence for one query protein (A or B above) fused with either the AD or the BD gene fragments. By simultaneously introducing the vectors for any pair of genes into the test system, one can quickly test whether the two proteins interact. This neatly reduces the $O(n^2)$ tasks into $O(n)$ slow steps (building the library) and $O(n^2)$ fast steps (doing individual tests).

[5]A vector is a DNA molecule that can be used to insert a user-specified DNA fragment into a target cell

2.5.2 Co-Immunoprecipitation

This method is more direct: break up the cell, collect all protein complexes containing a particular protein X, and then analyze these fragments to identify the other proteins in the accumulated complexes [28, 39, 59]. Here, the second part (collection) relies on a technique known as immunoprecipitation. A bait protein X is selected, with the only requirement being that antibodies that will bind to it should be available. These antibodies are coated onto a column of agarose beads (other materials may also be used). The cell extract is then passed through this column. The protein X and any proteins (preys) it is bound to get attached to the antibodies on the beads while everything else flows away. We then remove extraneous material by washing and then isolate the test material from the beads. The remaining fragments now consist only of bait and bait-prey fragments. These fragments are then analyzed by a mass spectrometric analysis to determine their chemical composition. By matching to the database of known protein sequences, one can then estimate which proteins are part of the complex.

The two techniques, Two-Hybrid (2H) and Co-Immunoprecipitation (Co-IP), each have their advantages. It may be argued that Co-IP is the more direct assay in that one actually extracts the protein complexes and analyzes them; in contrast, the 2H approach can lead to false-positives where the reporter gene gets activated even though the two proteins A and B would not interact under normal conditions (in Chapter 4 we address precisely this problem). On the other hand, Co-IP techniques require that a suitable antibody be available for the bait protein. They are also ill-suited to discover interactions that do not result in the formation of stable complexes. The extreme case of such interactions would be transient interactions between two proteins. It is unlikely that even 2H protocol is able to identify all such transient interactions; however, it is likely to be more powerful than the Co-IP technique in this regard.

A significant part of this thesis is focused on algorithms for correcting and complementing the data from these approaches. In Chapter 3, we describe a framework for predicting protein interactions, with one of the motivations being to identify interactions that the current experimental protocols can not. In Chapter 4, we describe an error-model specifically designed to address false-positives from 2H data.

2.6 Biological Datasets

In this section, I briefly list out a selected set of web-services and web-databases that provide biological data. The list is far from comprehensive; it is only intended to outline some of the core datasets that I found

useful in my research. Many of these are also good starting points for a computer scientist interested in playing with biological data. For more datasets, a particularly good compendium is in the Database and Web-Servers special issues, published annually by Nucleic Acids Research [19].

- **GenBank:** Genetic sequence database, maintained by the National Institutes of Health (NIH), containing all publicly available DNA sequences. `http://www.ncbi.nlm.nih.gov/genbank`
- **Pubmed:** A database, also from NIH, of references and abstracts of papers on biological and medical topics. `http://www.ncbi.nlm.nih.gov/pubmed`
- **Ensembl:** A parallel service to GenBank from Europe. `http://www.ensembl.org`
- **BioMart:** A web-tool the exposes the Ensembl data. It is extremely useful for getting the various gene/protein synonyms along with their sequence data, as well as a lot of other information. `http://www.ensembl.org/biomart/martview`
- **BioGRID:** One of the more comprehensive databases of experimentally determined protein-protein interactions. `http://thebiogrid.org`
- **Database of Interacting Proteins (DIP):** Similar to BioGRID, with a significant overlap to it. `http://dip.doe-mbi.ucla.edu`
- **STRINGS:** A database of predicted and experimentally-verified interactions, covering both genetic and protein-protein interactions. `http://string-db.org`
- **Gene Ontology (GO) Database**: The GO Consortium has worked to define a standard set of terms that may be used to describe a gene's function, role and cellular location. Furthermore, these terms are represented as part of a directed acyclic graph, capturing semantic relationships between them. The web-database contains this graph as well as a mapping of genes/proteins to these terms. `http://www.geneontology.org`
- **Gene Expression Omnibus:** Database of gene expression data from a variety of experiments, curated and maintained by the NIH. `http://www.ncbi.nlm.nih.gov/geo`
- **Protein Data Bank (PDB):** The database of protein structures, both for individual proteins and multi-protein complexes `http://www.pdb.org/`

- **Structural Classification of Proteins (SCOP**: This database is best thought of as a companion to the PDB; it provides a hierarchical classification of proteins grouping them by their structural similarity. `http://scop.mrc-lmb.cam.ac.uk/scop`
- **Homologene:** A database containing orthologs, i.e., sets of genes across various species that were derived from the same gene in a common ancestor. Orthologs often play similar roles in each species. `http://www.ncbi.nlm.nih.gov/homologene`
- **Inparanoid:** Another ortholog database, predicted using different techniques. `http://inparanoid.sbc.s`
- **FlyRNAi:** The Drosophila RNAi Screening Center contains a set of publicly available RNA-interference data from experiments on fruit-fly. `http://www.flyrnai.org/`

Chapter 3

Predicting Protein-Protein Interactions: Use of Structure-based Techniques

This chapter describes joint work with Jinbo Xu, Daniel Park and Bonnie Berger

3.1 Introduction

Until high-throughput techniques for discovering protein-protein interactions (PPIs) were invented, PPIs were largely the domain of the structural biologist. Traditionally, structural biologists have primarily been interested in understanding the *mechanism* of protein interaction. The motivation there has been to understand (1) how complex, multi-domain proteins are formed and, (2) how ligands (i.e. small molecules) bind to proteins. The latter goal is especially crucial in drug discovery. Clearly, this mechanism-oriented, "worm's-eye" view of protein interaction is quite distinct from the network-oriented, "bird's-eye" view emphasized in systems biology; the latter perspective emphasizes overall network properties over the mechanistic details of individual interactions.

This chapter aims to demonstrate how these perspectives can be combined. We describe a way of incorporating insights gleaned from structure-based approaches into a network-oriented analysis. Specifically, we use computational techniques inspired from structure-based analysis of protein interactions to make PPI predictions on a genome scale.

Species	Estimated Gene Count	Num Proteins With ≥ 1 PPI	(Percent)	Num Proteins With ≥ 5 PPIs	(Percent)
Saccharomyces cerevisiae	6275	5636	90%	4000	64%
Caenorhabditis elegans	20100	2862	14%	457	2%
Drosophila melanogaster	14000	7382	53%	2625	19%
Mus musculus	23000	1461	6%	274	1%
Homo sapiens	23000	9033	39%	3794	16%

Table 3.1: **Coverage of PPI data for the major model organisms**. Using data from BIOGRID [83] and estimates of gene counts for various species, we show here the relative coverage of PPIs for some of the major species of interest. Taken in conjunction with Fig 2-1, this suggests that while there has been a lot of increase in the amount of experimentally-determined PPI data, the coverage of PPI data still remains insufficient.

3.1.1 The Need To Predict Protein Interactions

Protein interactions can be discovered by a variety of techniques. An analysis of data from BIOGRID suggests that in the 1980's and 1990's, the most popular approaches to discover protein interactions were low-throughput approaches: co-purification, co-crystallization etc. Evaluating each putative interaction using these methods is a slow process. Over the last decade, however, the advent of high-throughput techniques has enabled genome-wide scans. The most popular among these Co-Immunoprecipitation and Yeast 2-Hybrid methods, both of which were briefly described in Section 2.5, The use of such high-throughput techniques has led to an explosion in the availability of protein interaction data over the last decade (see Fig 2-1).

Despite the advances in experimental techniques, the coverage of PPI data remains relatively low. One way of quantifying such coverage for a species is to count the number of proteins that have at least one experimentally known PPI. This is a generous way of quantifying coverage— after all, it is unlikely that a protein with just one known PPI has been fully "covered". Table 3.1 shows this coverage for some major species. Except for yeast, the coverage is about 50% (or much lower) for all the species. Furthermore, the growth-rate of PPI datasets' size has slowed in recent years (Fig 2-1). One of the problems in accumulating PPI data is that the set of possible proteins pairs to be investigated is extremely large, even with high-throughput methods: for example, in a species with about 23,000 genes (like in the human cell), the number of possible interactions is about 265 billion.

In this context, we believe that computational predictions of PPIs can be of significant value. First, such predictions can be used to shortlist potential PPIs that can be tested experimentally. Second, the ex-

perimental approaches are designed to discover only certain kinds of PPIs (or those involving certain kinds of proteins). For example, it has generally been difficult to experimentally discern PPIs involving transmembrane proteins. Also, current experimental techniques are biased towards discovering non-transient (i.e., complex-forming) interactions. Finally, high-throughput methods have a significant error (both false positives and false negatives); computational methods for predicting PPIs can be used to produce a confidence score for experimentally determined PPIs. Having a confidence score can be very useful in many network algorithms, which can then be tuned to make inferences that give greater weight to edges with higher confidence.

Existing work on predicting PPIs can be divided into two sets of approaches. The first set of approaches are model-driven, where the interaction between proteins is assumed to obey a certain abstract model. The second is a set of black-box, guilt-by-association approaches where non-PPI biological data is used to predict protein interactions. An example of model-based approaches are the methods proposed by Deng *et al.* [23] (and later, Wang *et al.* [91]): these posit that two proteins interact if they have a pair of compatible sequence domains. Each protein is modeled as a set of sequence domains and a pair of proteins are assumed to interact if and only if there is a pair of sequence domains (one from each protein) that interact. Given these models, it is easy to see how a machine learning algorithm can be trained to predict PPIs: given (1) a database that allows us to map a protein sequence to a set of sequence domains and, (2) a training set of positive (i.e., interactive protein pairs) as well as negative (i.e., non-interacting pairs) examples, one can estimate the most likely set of interacting domains that explain the training dataset. For any new protein pair, one can then simply check if it contains one of these interacting domain-pairs. If it does, the proteins are predicted to interact; otherwise, no interaction is predicted.

The insight guiding the guilt-by-association methods is that two proteins that interact are also likely to be co-localized, have similar functional annotation, and correspond to genes with similar expression profiles etc. Thus, there are a set of approaches that treat PPI prediction as a classical classification problem. The set of features can be rather broad (see Qi *et al.* [76] for a large list of these). The actual machine learning framework used for prediction also varies: Bayesian classifiers [51], Markov Random Fields [49], and support vector machines [40] are some of the approaches used.

In this chapter, we focus on prediction of protein interaction using structure based methods. Given a pair of protein sequences, we aim to predict the structure of the most likely joint-complex formed by them by using insights from structural biology literature. We then evaluate it using statistical mechanical energy functions and if the putative complex is sufficiently stable, we predict that the two proteins interact.

Unlike many other methods, our approach allows us to make predictions about proteins for which very little functional annotation is available. Also, it goes beyond black-box approaches by providing a model for *how* a pair of proteins interact, not just *if* they interact. We also describe a framework to integrate, using random forests, the predictions of this purely structure-based approach with functional genomic data like co-expression, co-localization, functional annotation etc. This allows us to build on the significant amount of previous work on predicting PPIs using machine learning approaches.

The rest of the chapter is organized as follows. We then formulate the problem with two cases: (1) when only the structure-based approach is used and (2) when this approach is combined with functional genomic data. Before describing the algorithm, we briefly review how structural biologists have typically aimed to understand protein interaction. After that, we describe the algorithm. We discuss some of the specific design choices made and how the various parameters are fitted. The next section discusses the algorithm's evaluation, starting from construction of training and test datasets to the performance of the algorithm. We follow that with a discussion of the benefits and drawbacks of the algorithm, especially in comparison to other approaches. Finally, we describe a web-service that allows users to provide a pair of proteins and query if an interaction is predicted between them.

3.2 Problem Formulation

We first consider the case when only the structure-based approach is used:

Problem [STRUCTONLY] Given two proteins p and q, their sequences S_p and S_q, and a database of protein-complex templates, compute the probability that p and q interact. We construct this database using information from SCOP [3] and PDB [11].

We now consider the more general case, where the structure-based approach is integrated with functional genomic data.

Problem [STRUCT&OTHERINFO] Here, we augment the previous problem with additional information. Given two proteins, p and q, their sequences S_p and S_q, and optional annotation information $\{X_p^1, X_p^2, \ldots\}$ and $\{X_q^1, X_q^2, \ldots\}$, compute the probability that p and q interact.

In STRUCTONLY, note that we only require the protein sequences, and not structures. If necessary, the protein sequences can themselves be inferred from the corresponding gene sequences. In STRUCT&OTHERINFO, different kinds of annotation information can be incorporated, as available. Our method for solving this

problem can be used with as many information sources as desired, but here we have restricted ourselves to a few information sources (see Table 3.2)

#	Name	Description
1	Coexpression	Similarity between expression levels of the corresponding genes
2	Colocalization	Co-localization information for the two proteins
3	GO	Similarity between Gene Ontology(GO) terms for the two genes
4	MIPS	Similarity between MIPS [66] terms for the corresponding genes
5	Domain	Seq. motifs indicating the presence of interacting domains
6	Coessentiality	Whether one, both, or none of the corresponding genes are *essential*

Table 3.2: **The various kinds of functional annotation used in STRUCT&OTHERINFO** . These benchmark annotations have previously been found to be particularly relevant in PPI predictions [76].

3.3 Algorithm

3.3.1 The Structural Biology of Protein Interaction

Structural biologists have long been interested in protein interaction. Their general goal has been to understand PPI from a *worm's eye* perspective: the mechanism of interaction between two proteins (or a protein and a ligand) and the principles underlying the interaction process. Most approaches towards solving this problem use computational models of protein structures to investigate and simulate the positioning, relative orientation and binding of proteins during the interaction process. In this chapter, we use some of the ideas from this field to develop algorithms for predicting a genome-scale protein interaction network. Below, we briefly review some of the popular approaches in this domain.

Most computational approaches to modeling interaction between two structures share a common assumption: the true joint structure of the two interacting proteins is the lowest energy conformation among all such possible conformations. Most algorithms for finding the joint structure then reduce to solving an optimization problem: that of finding the lowest energy conformation. The variations across algorithms lie in the kind of the optimization framework constructed and the solution techniques adopted. On one end of the spectrum are *ab-initio* methods. Here, the two proteins are represented by all-atom 3-D models, the search space includes all possible joint conformations and the search is guided by an all-atom energy function. This approach bears many similarities to the *ab-initio* protein structure prediction algorithms

and many of the terms in the energy functions are also similar (e.g. vanDer Waals interaction, hydrogen bonding etc.).

Further along the spectrum of approaches are methods that make *a priori* assumptions about the possible conformations of the joint structure, thus limiting the search space. An early example of this approach were models for analyzing coiled-coil proteins, where two or more alpha-helices wind around each other. This structure is characterized by heptad repeats and the amino-acids along the heptad govern the interaction specificity, e.g., between the various bZIP proteins [34]. Another set of approaches borrows ideas from the protein-threading literature. Here, the search space is constrained *a priori* by assuming that the joint conformation is similar to some known protein-complex's structure. The Protein Data Bank contains not just structures of single proteins but also structures of protein complexes. There are about 1200 such structures. The threading-based methods identify the template most likely to match the given protein pair and then the search for the optimal conformation is limited to conformations similar to the template.

As might be expected, these approaches embody different trade-offs in solving the underlying optimization problem. A key advantage of the *ab-initio* approach is that it starts from first principles and requires no information about the proteins apart from their structures. As such, it might be the only available approach for proteins on which other methods can not be applied due to the lack of matching templates. However, *ab-initio* approaches can be extremely computationally intensive. Furthermore, while the current methods provide good results when modeling interactions between proteins and small-molecules (the typical use-case in drug discovery), their accuracy declines markedly when modeling interaction between large proteins. Another issue with these approaches is the need for protein structure information. In the majority of cases where the structure has not been experimentally determined, the only way to use this approach is to employ computationally predicted structures of the protein pair. On the other hand, the advantage of a threading-based approach is that it usually works even for large proteins, as long as we can find suitable templates to guide the search. The latter requirement is usually the stumbling block in using these methods. Often, a good template may not be available, thus limiting the coverage of threading based approaches. The choice between these approaches depends on the task at hand. The *ab-initio* approach has been useful for understanding interaction between proteins and small-molecules. For our purposes, however, it was not very useful: its results when modeling interactions between two large proteins (the typical scenario in our analysis) were significantly worse that the threading-based approach's.

3.3.2 Algorithm for StructOnly

Our algorithm for the StructOnly problem consists of two stages. Given two protein sequences, in the first stage we *assume* that the two proteins form a complex and compute the corresponding interaction energy. Here, we exploit homology between the given protein pair and complexes with known structure. In the second stage, we use logistic regression to identify those pairs for which the interaction energy is low enough and, hence, an interaction is likely.

Stage 1: Computing The Most Likely Protein Complex

We first compute the most-likely structure of complex formed by the given protein pair assuming they do interact. There are two kinds of approaches one can take to predicting the structure of the complex: (1) predict the structure of the two proteins separately and then "dock" the two structures together, i.e., compute the lowest-energy joint conformation of the two structures, or (2) predict the structure of the joint complex directly from the sequences, by looking at known examples of protein complexes. Unfortunately, it turns out that the state-of-the-art in docking algorithms is not good enough for our purposes here. While current docking algorithms may have suitable performance when modeling interactions involving proteins and ligands (small molecules), their running time was prohibitively large when analyzing interactions between two proteins of moderate size and very often the docking algorithm could not converge to a solution. Overall, we found that the second approach was the better option.

To compute the putative complex and the corresponding energy, we introduce DblRap ("Double Raptor), a novel algorithm based on a protein threading approach. Threading approaches have been very successful in predicting the structure of individual proteins. There, these approaches start with a database of protein structure templates and given a protein sequence, attempt to find the structural template that best aligns with it. Then, the sequence is "threaded" onto the template, i.e., the residues of the sequence are aligned to the residues of the template (with possible gaps in each) such that energy of the new structure is minimized. We extend the general framework of protein threading to the case of analyzing pairs of proteins. Like the single-protein threading case, our algorithm for threading protein pairs also exploits the idea that if a pair of proteins interact in a specific way, their homologs will interact in a similar way.

We begin by constructing a database of templates, each template corresponding to the structure of a two-protein complex. The list of protein templates is derived by analyzing dimeric and multimeric structures from the PDB, filtered to remove templates that are more than 70% identical. We also use

SCOP [3] to further group the templates into distinct folds and remove certain redundant templates.

After constructing the complex template database, we then thread each sequence pair to all the templates in the database to find the best potential match. For each template complex, we construct an integer linear program (ILP) whose solution corresponds to the lowest energy threading of the input sequence pair onto the template. We search the entire template database to identify the template complex that has the best alignment with the input sequence pair. The ILP formulation of DblRap is based on the ILP formulation underlying Raptor [93], a program for predicting single-protein structure using threading. The Raptor formulation for constructing and solving the threading problem has proven to be quite powerful: Raptor won an award during CAFASP 2003 [94]. Here, we sketch out the constraints that are used to construct the ILP. For a more complete specification of the ILP, please see Xu *et al.* [93].

We first describe the ILP in the case of single-protein threading and then discuss how to extend it to the protein-complex case. Given a query sequence s of n residues that is to be threaded over a template t with m residues, our aim is to find the alignment of query residues with the template positions (allowing for gaps) that minimizes the energy of the template structure with the query residues in place of template residues. More formally, we begin by constructing L^2 binary variables where $L \leq n + m$ is the length of each sequence after adjusting for gaps. Each such binary variable v_{ij} is 1 if and only if the sequence position i aligns with the template position j. Either the sequence position or the template position (but not both) may correspond to a gap. We then set up a variety of constraints on these variables (introducing additional variables as necessary):

1. **Secondary structure conservation:** we model the template sequence as a sequence of cores joined by loops. Each core corresponds to a secondary structure element (e.g., α-helix or β-sheet). We require that all gaps in the template sequence be restricted to the loop regions between cores (and at each end of the structure). The assumption here is the secondary structure is conserved and that insertions and deletions happen in the loop regions.

2. **Self-consistency of the mapping:** we impose consistency constraints on the variables. At most one query residue can be aligned to a template position and vice-versa. Also, if a query residue s_i is aligned to a template position t_p, then a downstream query residue s_{i+a} can not be aligned to an earlier template position t_{p-b} where $a, b \geq 0$.

3. **Suitability of pairwise contacts:** One of the key factors impacting the alignment quality is how suitable the aligned query residues are for the pairwise contacts as defined by the template structure.

Here, a *contact* is defined to occur between two template positions if the distance between their Cα atoms is within $7\mathring{A}$ and they are at least 4 residues away from each other in the template sequence. Furthermore, we restrict the energy function to only analyze contacts between residues in the cores, i.e., we ignore contacts arising due to residues in the loops.

4. **Objective function:** this is a weighted sum of the following scores. The weights for these terms are fitted using a machine learning approach [93]:
   - an environment fitness score: how suitable each query residue is to the core it maybe part of
   - mutation score: how suitable each query residue is for the template position it is assigned to
   - secondary structure compatibility score: how suitable the query residues are to the secondary structure elements of the template structure
   - gap penalty
   - pairwise interaction score: this is based on evaluating the query residue-pairs in contact, as described above

The extension of this approach to the two-protein case is relatively straightforward. The structure template now corresponds to a two-protein complex, there is now a pair of query sequences, and the inter-residue contacts in the template are not just within each sub-structure but also between the two sub-structures (these contribute to the *interfacial energy*). Using this approach, for any given sequence pair (p and q), we generate two alignment scores (E_p, E_q), their associated z-scores (z_p, z_q) , alignment probabilities (P_p, P_q) and an interfacial energy (E_{pq}). These are fed into a logistic regression model to predict interaction.

Stage 2: From Energy Values to Interaction Probabilities

We use binary logistic regression [43] to classify whether a set of scores corresponds to an interaction or not. In binary logistic regression, the goal is to predict a binary output variable Y, given a set of r predictor variables $\mathbf{X} = \{X_1, X_2, \ldots, X_r\}$. For an instance i, suppose y_i and $\mathbf{x_i} = \{x_{i1}, x_{i2}, \ldots, x_{ir}\}$ are the random variables corresponding to Y and $\mathbf{X}$, respectively. Let $\theta_i = P(y_i = 1|\mathbf{x_i})$. In this model, the dependence of θ_i on $\mathbf{x_i}$ is expressed by the logit function:

$$logit(\theta_i) = log(\frac{\theta_i}{1-\theta_i}) = \alpha + \beta^t \mathbf{x_i} = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_r x_{ir} \text{ or} \tag{3.1}$$

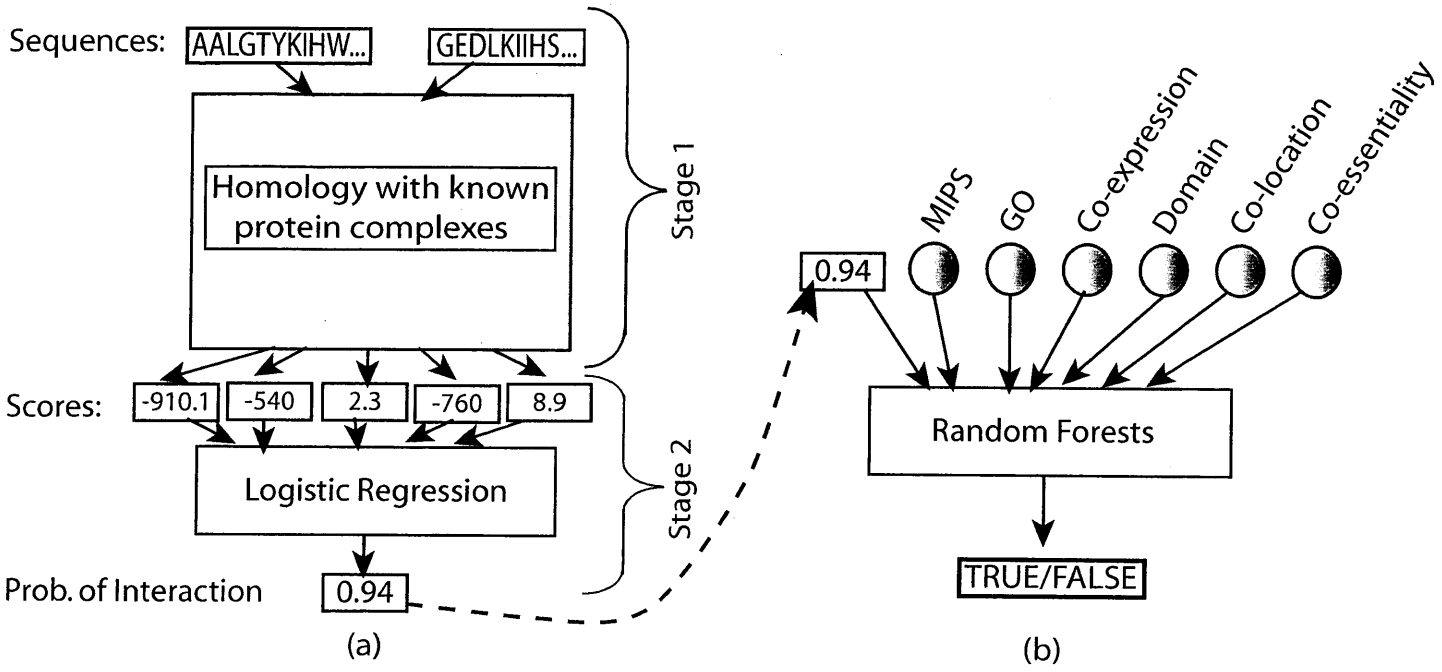


Figure 3-1: **Schematic of our method for (a) STRUCTONLY (b) STRUCT&OTHERINFO**

$$E(y_i) = \frac{1}{1 + \exp -(\alpha + \beta^t \mathbf{x_i})} \tag{3.2}$$

Logistic regression is a special case of a Generalized Linear Model. In such a model, the output y depends on a linear combination of the inputs x. The relation between the expected value $E(y)$ of the output variable and the linear combination of the inputs $\alpha + \beta^t \mathbf{x_i}$ is defined by a *link function*. Linear regression can be thought of as the special case where the link function is simply the identity function (i.e. $E(y) = \alpha + \beta^t \mathbf{x_i}$). In contrast, the link function in logistic regression it is the logistic function (3.2). Logistic regression is suitable for modeling cases where a binary-valued output variable depends on a combination of real-valued input variables. More precisely, the decision boundary (between positive and negative cases of the output) is assumed to be a straight line in the n-dimensional space of input variables. As a machine learning classifier, logistic regression provides us certain advantages. It is a simple model that avoids some of the overfitting pitfalls that more complicated approaches might have. Also, its predictions are real-valued scores that take values between 0 and 1. In contrast, other approaches (e.g. decision trees) would output just 1 or 0. The use of a real-valued score allows us to express confidence in our prediction and is particularly amenable to combination with functional genomic data (e.g., co-expression) in a larger classification framework.

In our case, the output variable Y is the probability of interaction of two proteins p and q. The predictor variables come from the first stage: the interfacial energy E_{pq}, the alignment scores E_p and E_q, as well as their associated z-scores (z_p, z_q) and alignment probabilities (P_p, P_q). In addition to these, we introduced additional predictor variables that normalize the energy values for protein size. Thus, $E_p/|s_p|, E_q/|s_q|$ are the energy scores for the two template sub-structures normalized by the respective sequence lengths. We intentionally built an initial model with an excessively large set of predictor variables: one of our goals was to identify the most informative subset of predictors. Towards this purposes, we performed stepwise variable addition/elimination in R, using the Akaike Information Criterion (AIC) at each step to evaluate whether the variable should be included in the model. The final set of selected variables makes intuitive sense: except for one, all of the energy terms enter only in their normalized, size-adjusted form:

- normalized alignment z-scores for the two sub-structures: $z_p/|s_p|$ and $z_q/|s_q|$
- alignment probabilities: P_p and P_q
- normalized alignment energy scores: $E_p/|s_p|$ and $E_q/|s_q|$
- raw interfacial energy: E_{pq}

To further gain confidence in our choice of selected predictor variables, we performed a similar feature selection using an alternative technique: L1 regularization. Regularization is the technique for adjusting a regression framework's objective function so that we minimize both model complexity as well prediction error. Plain vanilla logistic regression aims to find coefficents (represented here as a vector β) for the predictor variables such that the error in classification ε_β is minimized. In L1 regularization, a penalty term $\lambda|\beta|_1$ is added to the objective function. The intuition is that that the penalty term forces the aggregate weight of the coefficients $|\beta|_1$ to be small. As λ is increased from 0, the least useful predictor variables will successively drop out of the model (i.e., their coefficients β_i will become 0). Eventually, for high-enough λ, no variable will remain. This analysis generates what is called a *regularization path*: a sequence in which predictor variables should be added to the model, beginning with the most useful. We used R to compute the regularization path for our case. Here is the sequence of variables (in decreasing order of importance) as suggested by this analysis:

1. normalized alignment z-scores for the two sub-structures: $z_p/|s_p|$ and $z_q/|s_q|$
2. normalized alignment energy scores: $E_p/|s_p|$ and $E_q/|s_q|$

3. alignment probabilities: P_p and P_q

4. normalized interfacial energy: $E_{pq}/(|s_p| + |s_q|)$

5. raw interfacial energy: E_{pq}

6. raw alignment z-scores for the two sub-structures: z_p and z_q

7. raw alignment energy scores: E_p and E_q

Reassuringly, this mostly agrees with our original feature selection approach. If we were to only use the terms from Steps 1-4, our model from this approach would be very similar to the original model. The only variation would be that this model ranks normalized interfacial energy as slightly more useful than raw interfacial energy, in contrast to the original.

3.3.3 Algorithm for STRUCT&OTHERINFO

For classification purposes one can associate, with each pair of proteins p and q, a data-vector $D_{pq} = (d_1, \ldots, d_6)$ that contains information from the six non-structure-based information sources described in Table 3.2. To add structure-based information to this, we simply add one more feature d_7 to D_{pq}. Here, d_7 is the probability of interaction between proteins p and q as computed using logistic regression. Given some training data consisting of known true and likely false interactions, we then train a random forest to classify a possible interaction based on its data-vector (see Fig 3-1b).

Random forests [16] (RF) generalize the intuition behind decision trees, by employing ideas from bagging. They are an ensemble approach, like AdaBoost or bagging [26]. Instead of creating a single decision tree, in RF we create an ensemble of decision trees. To classify a point in the input space, a majority vote over the set of trees is used. Both the features used in each tree and subset of the training set used to construct it are randomly determined. This randomized approach has certain similarities to other ensemble approaches like bagging and boosting. Like bagging, each tree may be trained only on a subset of the training set. Like boosting, each tree may be trained using a subset of features. Unlike boosting, where feature-selection is guided by a deterministic weighting scheme (that emphasizes mis-classified examples), the features in each RF tree are randomly selected. Interestingly, Brieman has conjectured that in later stages of boosting, the deterministic approach might select features in a pseudo-random fashion, resulting in similar behavior to that in random forests.

Dataset	Interactions			Motivation behind creating the dataset	Post Filtering Interctns.
	Pos.	Notes	Neg.		
LT	100	From high-quality low-throughput experiments	400	Low-throughput interactions provide "gold-standard" positives	69
HTFEWANNOT	508	Between 1000 proteins with little functional annotation	2000	Existing guilt-by-association methods do not work well with these	332
HTMANYANNOT	489	Between proteins with a lot of functional annotation	300	Test how to combine structure-based methods with other info.	160

Table 3.3: **The construction of three datasets for yeast PPI data**. The positive interactions (#'s shown in table) were retrieved from BioGRID while (putative) negative interactions were generated by randomly pairing two yeast proteins. The difference between the datasets is primarily in how different positive sets were picked. The datasets were filtered to keep only those interactions for which homologous models could be found.

Random forests have many desirable characteristics. Like many other ensemble approaches, they are robust to overfitting errors. More importantly, even in cases where some of the training examples might be mislabeled, their performance does not degrade much. This is a significant advantage over a method like AdaBoost and is especially useful in our context. They also allow classification when features are not independent and have a good ability to estimate (and fill in) missing data. Thus, they are good at handling datasets with lots of missing values (again, a useful feature in this context). They are also useful in estimating the importance of many variables.

Our use of random forests is rather straightforward. Our feature space consists of the 7 features described earlier. We used the program written by Brieman and Cutler [16] to perform the training, the classification and the analysis.

3.4 Results

Datasets: We have focused our classification and evaluation analysis on predicting PPIs in yeast (*S. cerevisiae*) and fly (*D. melanogaster*), The list of experimentally discovered PPIs for these species was retrieved from BioGRID[83]. From this database, three datasets were created: LT, HTFEWANNOT, and HTMANYANNOT (see Table 3.3). The datasets differed in how their positive examples (true interactions) were selected (see Notes in Table 3.3). Note that because of the significant error-rate[90] in high-throughput experiments, some of the training data in HTFEWANNOT and HTMANYANNOT is likely to

be incorrectly labeled, i.e., some of the protein pairs in the positive dataset in these sets might not truly interact.

Our criteria for constructing positive and negative datasets were guided by the following intuitions:

1. Certain kinds of experiments are more reliable than others (e.g. co-crystallization experiments are likely more reliable than Yeast Two-Hybrid experiments)

2. Data from a paper publishing a very small number of datapoints is likely to be better validated than data from a paper with hundreds or thousands of observations

3. Previous research on the clustering characteristics of PPI networks has suggested that if protein pairs (A, B) and (B, C) are known to interact, then it is likely that the proteins A and C also interact.

We encode these intuitions as per the following criteria:

Positive Dataset: We aimed to identify the class of high-confidence, low-throughput experimental techniques in the BioGRID database. For this, we excluded techniques like affinity capture, two-hybrid, and those based on phenotypic activation/suppression or synthetic interaction. That left a set of experimental protocols that we deemed high-confidence. The most common remaining techniques were: reconstituted complex, biochemical activity, and dosage rescue. Typically, there were only a few interactions per publication for these experiments, further suggesting that these are low-throughput experiments. We included all interactions from these experiments.

We also included all interactions from papers where the published dataset has 5 or less reported PPIs. The intuition here is that the PPIs will be better validated in such papers than in papers with much larger-scale scans. Additionally, we included all reported PPIs such that the interacting pair (A, B) was connected by another protein C as well, i.e., there also existed PPIs (A, C) and (B, C).

Negative Dataset: In literature, it is difficult to find conclusive experimental data that some pair of proteins do not interact. Much of the previous work in PPI prediction has constructed negative training/test sets by using random pairs of proteins (and excluding those with a known interaction) [76]. The argument here is that the likelihood of interaction of a random pair of proteins is very small so it is reasonable to treat a random pair as a negative example of PPI. We chose a stricter version of this approach: we required the chosen (randomly selected) pair of proteins to either be disconnected in the experimentally-determined PPI network or be at least 3 hops away from each other in it. Essentially, we require that the two proteins not be co-clustered in the PPI network.

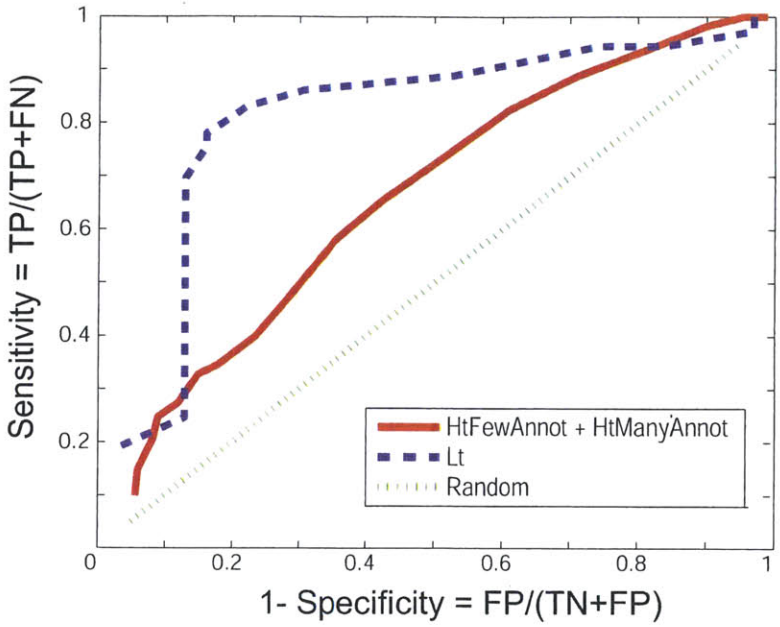


Figure 3-2: **Specificity-vs.-Sensitivity curve when using only the structure-based approach.** TP=True Pos., FP=False Pos., TN=True Neg., FN=False Neg. The dotted diagonal line indicates the baseline, a method with zero predictive power. The performance of our method is better for LT than for HTFEWANNOT +HTMANYANNOT. A possible reason might be that the latter datasets themselves might have mislabeled instances.

However, not all interactions in the datasets corresponded to protein pairs for which homologous complexes could be found. Therefore, we had to filter out a subset of the dataset. As discussed before, as more structures become available, the coverage of the homology-based methods will increase and fewer pairs will be filtered out.

Using Only Structure-based Method (STRUCTONLY): We tested our method by 4-fold cross-validation on the LT dataset. In addition, the method was trained on the entire LT dataset and tested on the combined HTFEWANNOT + HTMANYANNOT dataset. By comparing against some threshold value (say $p_{thresh} = 0.5$), the probabilities of interaction predicted by logistic regression can be interpreted as true/-false interactions. By varying p_{thresh}, we can plot the sensitivity-vs.-specificity (ROC) curve of the method (see Fig 3-2). As can be seen, the structure-based method provides significant signal for prediction purposes. The performance of the method is better on the low-throughput (LT) dataset than on the high-throughput datasets. A possible cause might be that the high-throughput datasets have more errors, i.e., negative examples mis-labeled as positive.

Combining Various Information Sources (STRUCT&OTHERINFO): We tested our entire framework on the HTMANYANNOT dataset, a dataset specifically chosen for proteins with lots of functional annotation available. We used 5-fold cross-validation to evaluate our method, using the cross-validation error (CVE) as the quality metric.[1]

With average sensitivity = 94.1% and specificity = 92.1%, the overall performance of our method is better than that of existing work, e.g., Zhang *et al.*'s[99] (sensitivity = 81% at specificity = 80%, approximately)[2]. Even when experimental PPI data itself has been used as one of the predictors by others (e.g., Lin *et al.* [61]: sensitivity = 98%, specificity = 92%, approximately), our method — which is *completely* independent of experimental PPI information — performs comparably.

3.5 Struct2Net Web-Server

To make the predictions of our approach widely available, we have created Struct2Net (http://struct2net.csail.mit.edu). It is a web-service that predicts interactions between proteins using a purely structure-based approach. We believe that it is the first community-wide resource to provide structure-based PPI predictions that go beyond homology modeling. Currently, most web-resources that provide computationally predicted PPIs (e.g., STRING[84]) rely on using functional genomic data (e.g., GO annotation, gene expression, cellular localization, etc) to make predictions. Our structure-based method is completely independent of such approaches; it can thus provide new information about the likelihood of a given protein-protein interaction. The displayed output includes the logistic regression score, allowing users to further filter the algorithm's predictions.

A fundamental trade-off with web-servers (especially, predictive ones) is between speed and quality: waiting for the results can be frustrating for some users who would prefer a quick (albeit, approximate) answer; for others, the quality of the produced predictions is paramount, even if the response is slow. With Struct2Net, we have strived to achieve a good balance between the two scenarios, and have aimed to provide full flexibility to the user. For the most commonly studied organisms (fly, human and yeast), we have precomputed all-vs.-all predictions and stored them. Users can retrieve these nearly instantaneously.

[1]Computing 5-fold cross-validation error (CVE): data was randomly partitioned into five equal parts. Four of the parts constituted the training set while the fifth one made up the test set. The error was computed as the classification error on this test set. By repeating this error computation for each of the classes, five error values were computed and averaged to compute the CVE.

[2]We compared against Zhang *et al.*'s performance in the case when they did not use experimental PPI data as a predictor

For proteins from other species, if the user desires a quick response of reasonable quality, we find the orthologs of the given protein(s) in the stored-set of yeast/human/fly proteins, map back the corresponding set of stored predictions to the given protein(s), and output this result. Finally, we provide the option of performing a full-blown prediction (which involves a threading algorithm and a machine learning algorithm); the user is emailed when the results for this are available.

We believe that this web-server is the first of its kind and will be of value to systems biologists interested in PPIs. Its predictions may be used by themselves or as one of the inputs into a computational framework that combines it with other sources (e.g. low-quality experimental data or predictions from functional genomic data).

3.6 Conclusion

We have described how structure-based methods can be integrated with other genomic and proteomic information for predicting PPIs. Structure-based methods can be used by themselves when other functional annotation is not available. When used in conjunction with functional annotation, their addition improves prediction accuracy over existing methods. A possible concern might be that current structure prediction methods are not sufficiently accurate and may not work well for every protein pair. In response, we note that our framework is modular so that better methods can be substituted in, as they become available. Second, our method is homology-based and will improve in performance and coverage as the recent NIH-funded push to elucidate more structures gains momentum.

Another concern might be that just because two protein structures interact *in-silico*, they might not interact *in-vivo*. This risk can be mitigated by combining inferences based on structural-techniques with other kinds of data. Also, note that this concern is equally applicable to existing approaches. Similarly, like many previous approaches, we restrict ourselves to pairwise protein interactions, even though more than two proteins may simultaneously interact *in vivo*.

Chapter 4

Modeling Systematic Errors in Yeast Two-Hybrid Data

This chapter describes joint work with David Sontag and Bonnie Berger.

4.1 Introduction

One of the issues associated with any high-throughput experimental approach is dealing with the errors, both random as well as systematic, associated with the process. Addressing these errors requires either changes in the protocol design, or a post-processing computational analysis or, often, both. Sometimes, the protocol is designed specifically to provide redundant observations or control data that make computational post-processing easier. For example, during the Human Genome Project, the genome was sequenced at approximately 12x coverage, i.e., each position in the genome was covered by about 12 independent, overlapping sequence fragments. This was to enable the sequence assembly software to robustly stitch together the true sequence while allowing for sequencing errors in individual fragments.

Here, we focus on methods for modeling and mitigating errors in high-throughput methods for discovering PPIs. The two commonly used approaches, Yeast 2-Hybrid (Y2H) and Co-Immunoprecipitation (CoIP) are both liable to random as well as systematic errors.

In this chapter, we aim to improve the quality of experimentally available PPI data by identifying erroneous datapoints from certain PPI experiments. We specifically focus on data from *Yeast Two-Hybrid* (YTH) experiments [48, 87], which are one of the most popular high-throughput methods for elucidating

protein-protein interaction. Data from YTH experiments forms a large fraction of the known PPI data for many species: *D. melanogaster, C. elegans, H. sapiens* etc. However, currently available YTH data also has unacceptably high false-positive rates: von Mering *et al.* estimate that more than 50% of the reported interaction in the early YTH interactions were spurious [90]. These high rates of error seriously hamper the ability to perform analyses of the PPI data. As such, an error model that performs better than existing models — even if it is tailored to YTH data — is of significant practical value, and may also serve as an example for the development of error models for other biological experiments.

4.1.1 Error Modeling in PPI Experiments

Previous computational methods of modeling systematic errors in PPI data can be broadly classified into two categories. The first class of methods [51, 90, 49] exploits the observation that if two very different experimental setups (e.g. YTH and Co-IP) observe a physical interaction, then the interaction is likely to be true. However, this approach requires many costly and time consuming genome-wide PPI experiments, and may still result in missed interactions, since the experiments have high false negative rates.

The second class of methods is based on the topological properties of the PPI networks. Bader *et al.*[4], in their pioneering work, used the number of YTH interactions per protein as a negative predictor of whether two proteins truly interact. Since the prior probability of any interaction is small, disproportionately many YTH interactions involving a particular protein could possibly be explained by it being self-activating or promiscuous. However, such an approach is unable to make fine-grained distinctions: an interaction involving a high-degree protein need not be incorrect, especially if there is support for it from other experiments. Furthermore, the high degree of a promiscuous protein in one experiment (e.g. Ito *et al.*'s) should not penalize interactions involving that protein observed in another experiment (e.g. Uetz *et al.*'s) if the errors are mostly independent (e.g. they use different reporters). Our proposed probabilistic models solve all of these problems.

The key contribution of this work is a comprehensive error model for YTH experiments that accounts for both random as well as systematic errors and is guided by insights into the systematic errors of the YTH experimental protocol. We believe this is the first model to account for both sources of error in a principled manner; in contrast, previous work on estimating error in PPI data has assumed that the error in YTH experiments (as in other experiments) is independent and random.

4.1.2 The Yeast Two Hybrid Protocol: Origins, Design, and Limitations

The Yeast Two Hybrid protocol was invented and described by Fields and Song in 1989 [31]. Although the initial work described the protocol for just a single yeast protein-pair, it quickly became clear that the protocol itself was much more generalizable, amenable to high-throughput approaches, and was useful for other species as well. Also, it is an *in vivo* approach, unlike many other approaches (e.g. co-purification or co-crystallization) which are *in vitro*. This is useful because the artificial conditions in an *in vitro* setup may distort the experimental results. At the time when the YTH protocol was invented, there were no other approaches for detecting PPIs that were nearly as powerful and efficient. As such, the advent of YTH marked a significant advance in the ability to investigate and analyze protein interactions. Furthermore, the protocol was shown to have utility beyond just protein-protein interactions: it has also been used for DNA-protein, RNA-protein and protein-ligand interactions.

The basic insight driving the YTH protocol is simple yet elegant. It emerged from an understanding of how transcription factors (e.g. *Gal4p* in yeast) function. A transcription factor typically activates its target gene by binding to the latter's upstream activating DNA sequence (UAS). In many transcription factors, the DNA-binding (DB) domain and the activation domain (AD) (i.e., the part responsible for activating the target gene) are in structurally separable parts of the molecule. The key insight in YTH protocol is to actually separate out the two domains and fuse them into two different proteins. If the two proteins interact, the DB and AD domains will be able to function in sync. Thus, the combined entity will successfully bind to the UAS region of the target gene and activate it. The target gene is typically a reporter gene (e.g., *lacZ*) whose activity can be easily measured using some well-known methods. The protocol lends itself to genome-scale analysis by pre-constructing libraries of DB-domain-fused genes ("bait" libraries) and AD-domain-fused genes ("preys"). These can then be crossed in an all-vs-all setup.

Like most experimental protocols, YTH experiments can give erroneous results, producing both false positives and false negatives. While there always are random errors (as in most experiments), the protocol itself is particularly susceptible to certain kinds of errors. For example, membrane proteins can not be easily localized to nucleus, a crucial requirement in YTH's transcription-based approach. This causes systematic false negatives related to interactions involving such proteins. On the other hand, false positives can be also occur in a systematic way in YTH experiments. There are two main ways such false positives occur. The first case is where the proteins interact in the experimental setup but do not actually interact inside the cell's natural environment (e.g., because of differing localization, expression profiles). The

second, and more frequent problem with YTH [89], is that certain proteins can trigger the expression of the reporter gene independently of any protein-protein interaction. These may be proteins that can activate transcription by themselves when bound to the DB-domain or the AD-domain. Alternatively, they may be involved in the constitutive expression of the reporter gene. In YTH output, these proteins show up repeatedly (i.e., in multiple PPIs), displaying what has been called *promiscuous* binding. Vidalain *et al.* have and described some changes in the experimental setup to reduce the problem [89]. Our work aims to provide a parallel, computational model of the problem, allowing post-facto filtering of data, even if the original experiment retained the errors.

4.2 Probabilistic Modeling of Yeast Two-Hybrid Errors

We use the framework of Bayesian networks to encode our assumption that a YTH interaction is likely to be observed if the corresponding protein pair truly interacts or if either of the proteins is self-activating/promiscuous. The Bayesian framework allows us to represent the inherent uncertainty and the relationship between promiscuity of proteins, true interactions and observed YTH data, while using all the data available to simultaneously learn the model parameters and predict the interactions. We use a Markov Chain Monte Carlo (MCMC) algorithm to do approximate probabilistic inference in our models, jointly inferring both desired sets of quantities: the probability of interaction, and the propensity of a protein for self-activation/promiscuity.

Our models can also adjust to varying error rates in different experiments. For instance, while we account for random noise and false negatives in our error model for data from both Uetz *et al.* [87] (the UETZ2H dataset) and Ito *et al.* [48] (the ITO2H dataset), we only model self-activation/promiscuity for ITO2H observations. The UETZ2H data set was smaller and included only one protein with degree larger than 20; ITO2H had 36 proteins with degree larger than 30, one with degree as high as 285. Thus, while modeling promiscuity made a big difference for the ITO2H data, it did not significantly affect our results on the UETZ2H data.

4.2.1 Generative model

We begin by describing a novel generative model in which the self-activating/promiscuous tendencies of particular proteins are explicitly modeled. We represent the uncertainty about a protein interaction

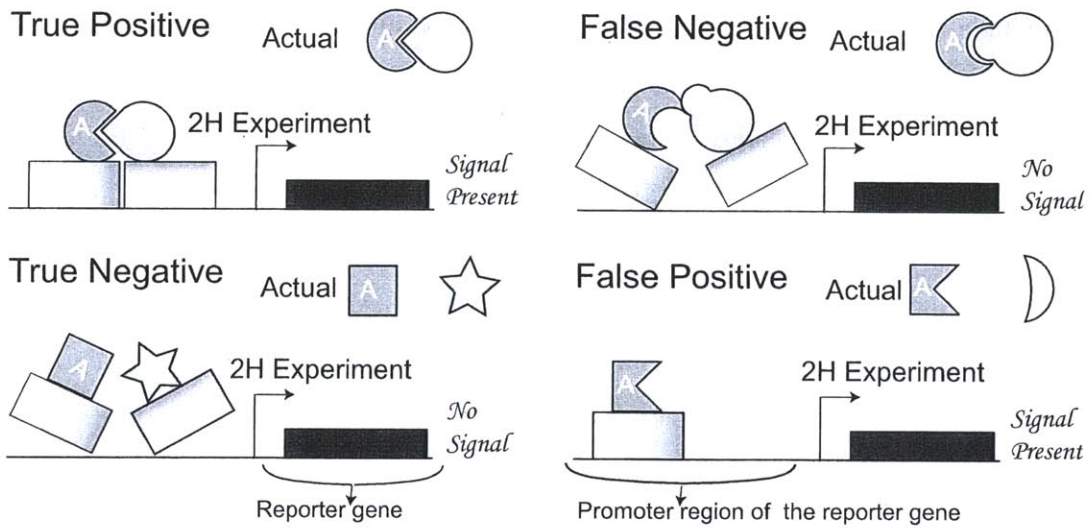


Figure 4-1: **The origin of systematic errors in YTH data.** The cartoons shown above demonstrate the mechanism of YTH experiments. Protein A is fused to the DNA binding domain of a particular transcription factor, while protein B is fused to the activation domain of that transcription factor. If A and B physically interact then the combined influence of their respective enhancers results in the activation of the reporter gene. Systematic errors in such experiments may arise: false negatives occur when two proteins which interact *in-vivo* fail to activate the reporter gene under experimental conditions. False positives may occur due to proteins which trigger the reporting mechanism of the system, either by themselves (self-activation) or by spurious interaction with other proteins (promiscuity). Spurious interaction can occur when a protein is grossly over-expressed. In the above figure, protein A in the lower right panel is such a protein: it may either promiscuously bind with B or activate the reporting mechanism even in the absence of B.

as an indicator random variable X_{ij}, which is 1 if proteins i and j truly interact, and 0 otherwise. For each experiment, we construct corresponding random variables (RVs) indicating if i and j have been observed to interact under that experiment. Thus, U_{ij} is the observed[1] random variable (RV) representing the observation from UETZ2H, and I_{ij} is the observed RV representing the observation from ITO2H. The arrow from X_{ij} to I_{ij} indicates the dependency of I_{ij} on X_{ij}. The latent Bernoulli RV F_k is 1 if protein k is believed to be promiscuous or self-activating. In the context of our data set, this RV applies specifically to the ITO2H data; if self-activation/promiscuity in multiple experiments is to be modeled, we may introduce multiple such variables F_k^H (for protein k and experiment H). The I_{ij} RV thus depends on F_i and F_j. Intuitively, I_{ij} will be > 0 if either $X_{ij} = 1$ or $F_k = 1$. As we show later in the Results section, this model of noise is significantly more powerful than the earlier model, because it allows for the "explaining away" of false positives in ITO2H. Furthermore, it allows evidence from data sets other than ITO2H to influence (through the X_{ij} RVs) the determination of the F_k RVs. We also added the latent variables O_{ij}^U and O_{ij}^I, which will be 1 if the Uetz *et al.* and Ito *et al.* experiments, respectively, have the capacity to observe a possible interaction between proteins i and j. These RVs act to explain away the false negatives in UETZ2H and ITO2H. We believe that these RVs will be particularly useful for species where we have relatively little PPI data. The distributions in these models all have Dirichlet priors (θ) with associated hyperparameters α (see Supp. Info. for more details).

The model is called "generative" because the ground truth about the interaction, X_{ij}, generates the observations in the YTH experiments, I_{ij} and U_{ij}. Compared to previous generative models, our approach allows for more fine-tuned modeling of false positives and false negatives. To our knowledge, all previous generative models of experimental interactions allowed for false positives by saying that $Pr(I_{ij} > 0 | X_{ij} = 0) = \delta_{fp}$, where δ_{fp} is a parameter of their model. Similarly, they allowed for false negatives by saying that $Pr(I_{ij} = 0 | X_{ij} = 1) = \delta_{fn}$, for another parameter δ_{fn}. However, these models are missing much of the picture. For example, many experiments have particular difficulty testing the interactions of proteins along the membrane. For these proteins, δ_{fn} should be significantly higher. In the YTH experiment, for interactions that involve self-activating/promiscuous proteins, δ_{fp} will be significantly higher. Our approach allows for such variations.

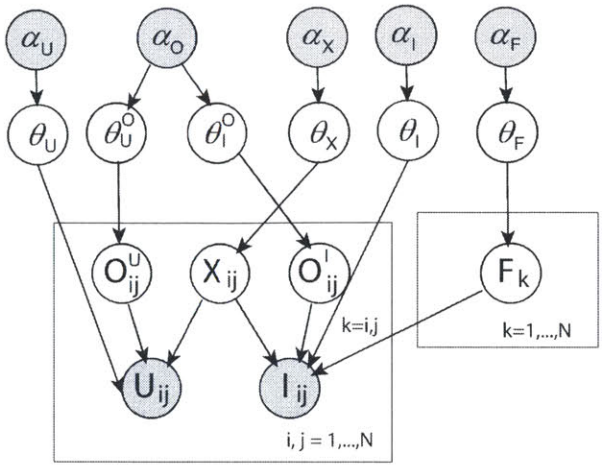


Figure 4-2: **Generative model, with noise variables**

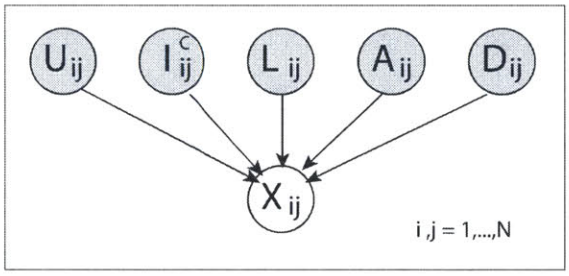


Figure 4-3: **Bader *et al.*'s logistic regression model (BADERLR)**

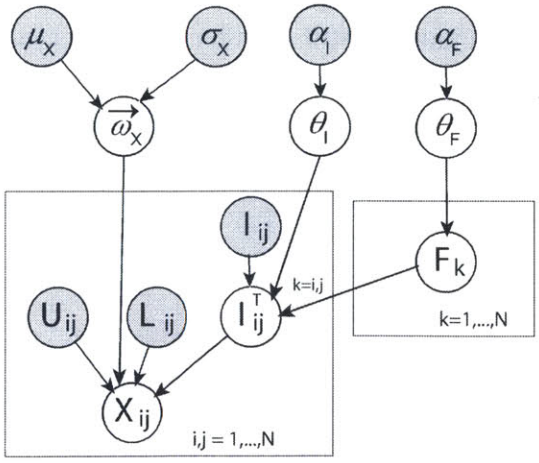


Figure 4-4: **Our Bayesian logistic model, with noise variables (BAYESLR)**

4.2.2 Bayesian logistic model

In Fig. 4-3 we show Bader *et al.*'s model (BADERLR); it includes three new variables in addition to some of the RVs already mentioned, whose values are pre-calculated using the YTH network. Two of these encode topological information: variable A_{ij} is the number of adjacent proteins in common between i and j, and variable D_{ij} is $\ln(d_i+1)+\ln(d_j+1)$, where d_i is the degree of protein i. Variable L_{ij} is an indicator variable for whether this protein interaction has been observed in any low-throughput experiments. In Bader *et al.*'s model, I_{ij}^C is an indicator variable representing whether the interaction between proteins i and j was in the ITOCORE data set (IST $\geq$ 3). X_{ij}'s conditional distribution is given by the logistic function:

$$p(X_{ij}=1) = \frac{1}{1+\exp\left(-(w_{offset}+U_{ij}w_U+I_{ij}^C w_I+L_{ij}w_L+A_{ij}w_A+D_{ij}w_D)\right)}.$$

The weights w are discriminatively learned using the Iterative Re-weighted Least Squares (IRLS) algorithm, which requires that all of the above quantities are observed in the training data.

In Fig. 4-4 we propose a new model (BAYESLR), with two significant differences. First, we no longer use the two proteins' degree, D_{ij}, and instead integrate our noise model in the form of the F_k random variables. Second, instead of learning the model using IRLS, we assign the weights uninformative priors and do inference via Markov Chain Monte Carlo (MCMC) methods. This will be necessary because X_{ij} will have an unobserved parent, I_{ij}^T. The new RV I_{ij}^T will be 1 when the Ito *et al.* experiment should be considered for predicting X_{ij}. Intuitively, its value should be $(I_{ij} > 0) \wedge \neg(F_i \vee F_j)$. However, to allow greater flexibility, we give the conditional distribution for I_{ij}^T a Dirichlet prior, resulting in a noisy version of the above logical expression. The RVs O_{ij} are not needed in this logistic model because the parameterization of the X_{ij} conditional distribution induces a type of noisy OR distribution in the posterior. Thus, logistic models can easily handle false negatives.

4.2.3 Inference

As is common in probabilistic relational models, the parameters for the conditional distributions of each RV are shared across all of their instances. For example, in the generative model, the prior probability $Pr(X_{ij} = 1)$ is the same for all i and j. With the exception of X_{ij} in BAYESLR, we gave all the

[1] Clear nodes are unobserved (latent) RVs, and shaded nodes are observed RVs.

distributions a Dirichlet prior. In BAYESLR, the conditional distribution of X_{ij} is the logistic function, and its weights are given uninformative Gaussian priors with mean $\mu_X = 0$ and variance $\sigma_X^2 = 100$. Note that by specifying these hyperparameters (e.g. μ_X, σ_X^2), we never need to do learning of the parameters (i.e., weights). Given the relational nature of our data, and the relatively small amount of it, we think that this Bayesian approach is well-suited. We prevent the models from growing too large by only including protein pairs where at least one experiment hinted at an interaction.

We used BUGS [62] to do inference via Gibbs sampling. We ran 12 MCMC chains for 6000 samples each, from which we computed the desired marginal posterior probabilities. The process is simple enough that someone without much knowledge of machine learning could take our probabilistic models (Tab 4.1, 4.2, 4.3) and use them to interpret the results of their YTH experiments.

4.3 Data Sets for Evaluation

We constructed a gold standard data set of protein-protein interactions in *S. cerevisiae* (yeast) from which we could validate our methods and compare the results to that of Bader *et al.* Our gold standard test set is an updated version of Bader *et al.*'s data. Bader *et al.*'s data consisted of all published interactions found by YTH experiments; data from experiments by Uetz *et al.* [87] (the UETZ2H data set) and Ito *et al.* [48] (the ITO2H data set) comprised the bulk of the data set. They also included as possible protein interactions all protein pairs that were of distance at most two in the YTH network. Bader *et al.* then used published Co-Immunoprecipitation (Co-IP) data to give labels to these purported interactions. When two proteins were found in a bait-hit or hit-hit interaction in Co-IP, they were labeled as having a true interaction. When two proteins were very far apart in the Co-IP network (distance larger than three), they were labeled as not interacting. We were able to update Bader *et al.*'s data with additional YTH interactions. Since the goal of our algorithms is to model the systematic errors specifically in YTH experiments, we evaluated our models' performance on the test data where at least one of UETZ2H or ITO2H indicated an interaction.

We were left with 397 positive examples, 2298 negative examples, and 2366 unlabeled interactions. We randomly chose 397 of the 2298 negative examples to be part of our test set. For all of the experiments we performed 4-fold cross validation on the test set, hiding one fourth of the labels while using the remaining labeled data during inference.

4.4 Results

We compared the proposed Bayesian logistic model (BAYESLR) with the model based on Bader *et al.*'s work (BADERLR). Both models were trained and tested on the new, updated version of Bader *et al.*'s gold standard data set. We show in Fig. 4-5 that BAYESLR achieves 5-10% higher accuracy at most points along the ROC curve. In all regimes of the ROC curve, BAYESLR performs at least as well as BADERLR; in some, it performs significantly better (Fig. 4-7). The examples that follow demonstrate the weaknesses inherent in BADERLR and show how the proposed model BAYESLR solves these problems. When IRLS learns the weight for the degree variable (in BADERLR), it must trade off having too high a weight, which would cause other features to be ignored, and having too low a weight, which would insufficiently penalize the false positives caused by self-activation/promiscuity. In BADERLR, a high degree D_{ij} penalizes positive predictors from all the experiments (U_{ij}, I_{ij}, L_{ij}). However, the degree of a protein in a particular experiment (say, Ito *et al.*'s) only gives information about self-activation/promiscuity of the protein in that experiment. Thus, if a protein has a high degree in one experiment, even if that experiment did not predict an interaction (involving some other protein), the degree will negatively affect any predictions made by other experiments on that protein. Our proposed models solve this problem by giving every experiment a different noise model, and by having each noise model be conditionally independent given the X_{ij} variables. Thus, we get the desired property that noise in one experiment should not affect the influence of other experiments on the X_{ij} variables.

Fig. 4-7(a) illustrates this by showing the prediction accuracy for the test points where $D_{ij} > 4$ and $U_{ij} = 1$ or $L_{ij} = 1$ (called the 'medium' degree range). When the degree of a protein is very high, BADERLR will always classify interactions involving it as false positives. Fig. 4-7(b) shows the setting of $D_{ij} > 6$[2]. With a false positive rate of less than 1%, BADERLR detects 42% of the true interactions, while BAYESLR detects 74% of the true interactions, a 76% improvement. Bader *et al.* found that they got better performance by using only a subset (where IST $\geq$ 3) of the interactions in ITO2H. Our noise model allows us to make use of all of the predicted interactions, without hurting our overall results. As a result, our predictions for the proteins pairs where Bader *et al.*'s model ignored ITO2H's interactions (i.e. IST $<$ 3) are highly more accurate. This is illustrated in Fig. 4-7(c). Finally, we show in Fig. 4-7(d) that at the very extreme when neither ITOCORE, nor the low-throughput YTH experiments (Lit), nor UETZ2H showed an interaction, we can still make meaningful predictions, using a combination of the noise model

[2]Recall that D_{ij} is on a log-scale, and is the sum for both proteins..

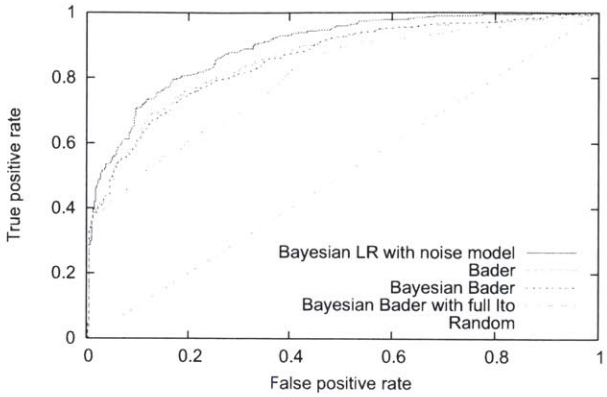


Figure 4-5: **Comparison of logistic models**

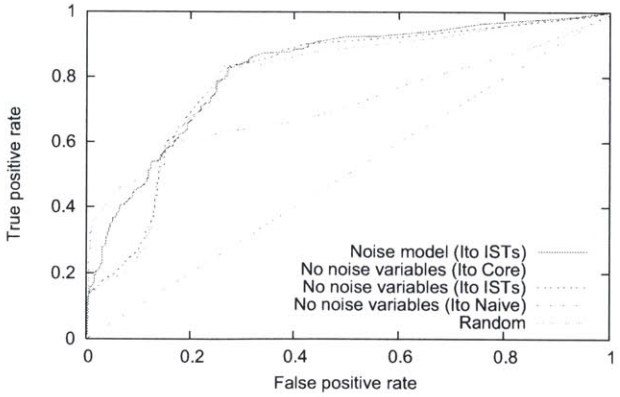


Figure 4-6: **Comparison of generative models**

and the observed interactions in Ito *et al.* where IST < 3.

We next compared the various generative models, with the results shown in Fig. 4-6. Naively implementing a simple random-error model (by omitting the O and F variables from Fig. 4-2) and by using an indicator variable for whether the interaction was observed in ITO2H, results in the worst performance. Changing the indicator variable to a discretized IST count significantly improves performance. Using our noise model (i.e. the model from Fig. 4-2) provides further improvements, especially in the lower left corner, where the previous two had performed poorly. However, if we simplify that generative model by removing the noise variables from that model and instead pre-filter the data as Bader *et al.* did, using an indicator variable for whether IST ≥ 3 in ITO2H, we performance that is almost as good. The noise model still does better in the upper half of the ROC curve, which is arguably where it matters the most. It is also interesting that our noise model is able to recover the accuracy of the hand-filtered IST≥ 3 criterion.

4.5 Conclusion

We have presented a principled approach to modeling the random and systematic sources of error in two-hybrid experiments, and showed how to integrate our noise models into the two most common probabilistic models for integrating PPI data. Comparisons with previous work demonstrate that explicit modeling of the sources of error can improve protein-protein interaction prediction, making better use of experimental data.

Future work could involve discriminative training of the generative models, investigation of systematic sources of noise in other biological experiments such as Co-IP, and applying noise models to the Markov networks of Jaimovich *et al.* [49] and possibly even in a first-order probabilistic model, where more intricate properties of proteins can be described and jointly predicted.

```
model {
  itodist[1,1,1,1:4] ~ ddirch(alphaito1[])
  itodist[1,1,2,1:4] ~ ddirch(alphaito2[])
  itodist[1,2,1,1:4] ~ ddirch(alphaito1[])
  itodist[1,2,2,1:4] ~ ddirch(alphaito2[])
  itodist[2,1,1,1:4] ~ ddirch(alphaito1[])
  itodist[2,1,2,1:4] ~ ddirch(alphaito2[])
  itodist[2,2,1,1:4] ~ ddirch(alphaito2[])
  itodist[2,2,2,1:4] ~ ddirch(alphaito2[])

  uetzdist[1,1,1:2] ~ ddirch(alphabin1[])
  uetzdist[1,2,1:2] ~ ddirch(alphabin1[])
  uetzdist[2,1,1:2] ~ ddirch(alphabin1[])
  uetzdist[2,2,1:2] ~ ddirch(alphabin2[])

  litdist[1,1,1:2] ~ ddirch(alphabin1[])
  litdist[1,2,1:2] ~ ddirch(alphabin1[])
  litdist[2,1,1:2] ~ ddirch(alphabin1[])
  litdist[2,2,1:2] ~ ddirch(alphabin2[])

  ppiprior[1:2] ~ ddirch(alphappi[])
  itofpprior[1:2] ~ ddirch(alphafp[])

  seeitoprior[1:2] ~ ddirch(alphasee[])
  seeuetzprior[1:2] ~ ddirch(alphasee[])
  seelitprior[1:2] ~ ddirch(alphasee[])

  for( p in 1 : N) {
       itopfp[p] ~ dcat(itofpprior[])
  }

  for( i in 1 : M ) {
       ppi[i] ~ dcat(ppiprior[])

       # Explaining away variable for ito,uetz,lit (if 0)
       seeito[i] ~ dcat(seeitoprior[])
       seeuetz[i] ~ dcat(seeuetzprior[])
       seelit[i] ~ dcat(seelitprior[])

       itofp[i] <- step(itopfp[parent1[i]] + itopfp[parent2[i]] - 3) + 1
       ito[i] ~ dcat(itodist[seeito[i],ppi[i],itofp[i],])
       uetz[i] ~ dcat(uetzdist[seeuetz[i],ppi[i],])
       lit[i] ~ dcat(litdist[seelit[i],ppi[i],])
  }
}
```

Table 4.1: **BUGS code for Generative Model in Fig 4-2**

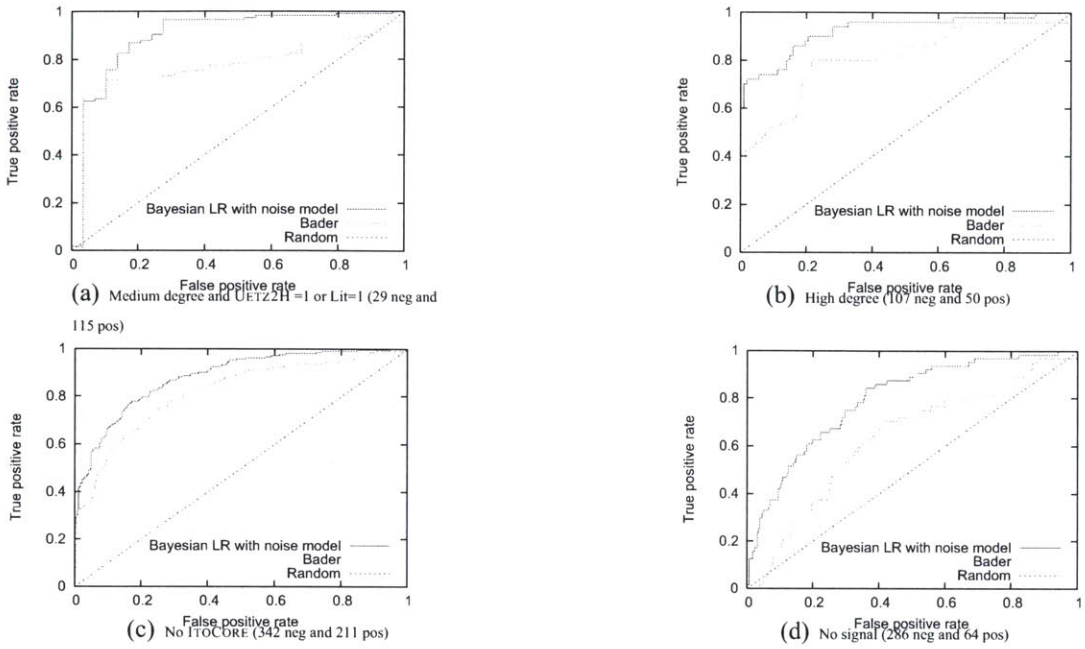


Figure 4-7: **Examples of regimes where the noise model is particularly helpful. In parentheses we give the number of test cases that fall into each category.**

```
model {

  # Uninformative priors for logistic regression weights
  logoffset ~ dnorm(0.0, 0.01)
  logdegree ~ dnorm(0.0, 0.1)
  loglit ~ dnorm(0.0, 0.01)
  logito ~ dnorm(0.0, 0.01)
  loguetz ~ dnorm(0.0, 0.01)

  for( i in 1 : M ) {
        val[i] <- logoffset + loglit*(lit[i]-1) + logito*(ito[i]-1) + loguetz*(uetz[i]-1) + logdc
        ppiprior[i] <- 1 / (1 + exp(val[i]))
        ppi[i] ~ dbern(ppiprior[i])
  }
}
```

Table 4.2: **BUGS code for Bader's Logistic Model in Fig 4-3**

```
model {

  itofpprior[1:2] ~ ddirch(alphafp[])

  # Uninformative priors for logistic regression weights
  logoffset ~ dnorm(0.0, 0.01)
  loglit ~ dnorm(0.0, 0.01)
  logito ~ dnorm(0.0, 0.01)
  loguetz ~ dnorm(0.0, 0.01)

  titodist[1,1,1:2] ~ ddirch(alphatito1[])
  titodist[1,2,1:2] ~ ddirch(alphatito1[])
  titodist[2,1,1:2] ~ ddirch(alphatito2[])
  titodist[2,2,1:2] ~ ddirch(alphatito1[])
  titodist[3,1,1:2] ~ ddirch(alphatito2[])
  titodist[3,2,1:2] ~ ddirch(alphatito1[])
  titodist[4,1,1:2] ~ ddirch(alphatito2[])
  titodist[4,2,1:2] ~ ddirch(alphatito1[])

  for( p in 1 : N) {
       itopfp[p] ~ dcat(itofpprior[])
  }

  for( i in 1 : M ) {
        itofp[i] <- step(itopfp[parent1[i]] + itopfp[parent2[i]] - 3) + 1
        tito[i] ~ dcat(titodist[ito[i],itofp[i],])

        val[i] <- logoffset + loglit*(lit[i]-1) + logito*(tito[i]-1) + loguetz*(uetz[i]-1)
        ppiprior[i] <- 1 / (1 + exp(val[i]))
        ppi[i] ~ dbern(ppiprior[i])
  }
}
```

Table 4.3: **BUGS code for Our Logistic Model in Fig 4-4**

Chapter 5

Comparative Analysis of Protein Interaction Networks

Section 5.2 describes joint work with Jinbo Xu and Bonnie Berger. Section 5.3 describes joint work with Chung-Shou Liao, Kanghao Lu, Michael Baym and Bonnie Berger

As the size of PPI datasets for various species rapidly increases, comparative analysis of PPI networks across species is proving to be a valuable tool. Such analysis is similar in spirit to traditional sequence-based comparative genomic analyses; it also promises commensurate insights. As a phylogenetic tool, it offers a function oriented perspective that complements traditional sequence-based methods. Comparative network analysis also enables us to identify conserved functional components across species [33] and perform high-quality ortholog prediction (i.e., identifying genes in different species derived from the same ancestral region). Solving these problems is crucial for transferring insights and information across species, allowing us to perform experiments in (say) yeast or fly and apply those insights toward understanding mechanisms of human diseases [88]. Indeed, Bandyopadhyay et al. [6] have demonstrated that the use of PPI networks in computing orthologs produces orthology mappings that better conserve protein function across species (i.e., functional orthologs).

One of the first comparative analyses of PPI networks was aimed at identifying the general network characteristics (e.g., the degree distribution, connectedness etc.) common across various PPI networks [52]. This analysis suggested that most PPI networks follow a scale-free topology: the degree distribution $f(d)$ of the nodes in these networks follows a power-law distribution, $f(d) \sim cd^{\lambda}$, where $f(d)$ is the frequency of nodes with degree d. Similar analyses have revealed the important roles that high-degree

proteins ("hubs") play in PPI networks. Most of the paths in a PPI network are routed through such hub proteins.

Another approach to comparing PPI networks has focused on the idea of network motifs: each such motif is a small graph (typically, with 4 or less nodes). Given a PPI network, we construct a motif-based signature of the network by enumerating how many times each motif occurs in the network. Such signatures do capture some information about the network: for example, if triangles are relatively more frequent than other 3-node motifs, it would suggest a high-degree of co-clustering in the network. In principle, these signatures can also be used to compare various PPI networks. However, such comparative analysis is not very fruitful. While the intuition behind network motifs is similar to the concept of sequence motifs popular in comparative sequence analysis, the former do not seem to be as useful as the latter. The key problem with the network motif approach is its weakens in summarizing PPI data. Sequence motifs have proven very useful because the fundamental characteristic of the underlying data – sequential ordering of nucleic-acids/amino-acids – is well captured by sequence motifs. Sequence motifs reduce a gene/protein sequence to its most-interesting segments. In contrast, the combinatorial nature of a large PPI network can not be easily captured by representing it as an unordered collection of many tiny graphs. A lot of biological detail is lost in the process.

Local alignment of PPI networks has been a particularly popular approach to comparative analysis of PPI networks. Here the goal is to find pathways and network-fragments common to two or more PPI networks. More precisely, the subgraphs from each network are required to be approximately isomorphic while the sets of corresponding nodes should be sequence-similar. In a variation of this approach, a query graph pattern is searched for in a given network. The pioneering work of Kelley et al. [54] described how BLAST similarity scores and PPI network information could be used to identify conserved functional motifs. Koyuturk et al. [58] proposed another method, motivated by biological models of duplication and deletion. Recently, Flannick et al. [33] proposed a new efficient approach, using modules of proteins to infer the alignment. Berg and Lassig [9] have proposed a Bayesian approach to this problem. Many of these methods limit the set of possible node-pairings based on sequence-based similarity scores or orthology predictions, and then add in network data to infer the alignment. This approach helps reduce the problem complexity, but lacks the flexibility of producing node-pairings that diverge from sequence-only predictions.

In this chapter, we introduce an approach to comparative analysis of PPI networks that addresses the problem of finding the optimal global alignment between two or more PPI networks. We propose the

ISORANK algorithm for multiple network alignment, aimed at finding a correspondence between nodes and edges of the input networks that maximizes the overall "match" between the networks. To the best of our knowledge, it is the first such algorithm of its kind. It simultaneously uses both PPI network data and sequence similarity data to compute the alignment, the relative weights of the two data sources being a free parameter. The algorithm is intuitive: a node i in G_1 is mapped to a node j in G_2 if the neighborhood topologies of i and j are similar, i.e., the neighbors of i can be well-mapped to the neighbors of j. This approach has parallels to Google's PageRank technique; like the latter, we formalize our intuition as an eigenvalue problem (see §5.2). ISORANK is, by design, tolerant to errors in the input (e.g., missing or spurious edges) and takes advantage of edge confidence scores as well as other biological signals (e.g. sequence similarity scores), when available. We use the algorithm to compute the first known global alignment of the most-commonly studies eukaryotic species:

We use ISORANK to simultaneously align the PPI networks of *Saccharomyces cerevisiae, Drosophila melanogaster, Caenorhabditis elegans, Mus musculus*, and *Homo sapiens*, the species that make up the bulk of available PPI data. The conserved subgraphs in this alignment are larger and more varied than those produced by previous methods, which performed pairwise network alignments. We also use the alignment results to predict functional orthologs across species and demonstrate that incorporating PPI data in ortholog prediction results in improvements over existing sequence-only approaches such as Homologene (`http://www.ncbi.nlm.nih.gov/homologene`) and Inparanoid [77]. Moreover, we find our pairwise alignment of yeast and fly networks produces functional orthology mappings that compare favorably with those from local alignments of the two networks. To test the biological quality of our predictions, we introduce a direct, automated method for scoring the quality of an ortholog list.

We note here that the graph alignment problem has also been studied in other domains. For example, in computer vision, the problem of matching a query image to an existing image in the database has often been formulated as a graph-matching problem, each image represented as a graph. Some of the solutions proposed in that domain use spectral techniques, i.e., they use eigenvalues computed based on each graph (14, 15). Our approach, which also constructs an eigenvalue problem (although, not for individual graphs) may be relevant in this domain as well.

5.0.1 Global vs. Local Network Alignment

In general, the goal in a network alignment problem is to find a common subgraph (i.e., a set of conserved edges) across the input networks. Corresponding to these conserved edges, there exists a mapping between the nodes of the networks. For example, when protein a_1 from network G_1 is mapped to proteins a_2 from G_2 and a_3 from G_3, then a_1, a_2, and a_3 refer to the same node in the set of conserved edges. What makes the problem difficult is the tradeoff involved: Maximizing the overlap between the networks (i.e., the number of conserved edges), while ensuring that the proteins mapped to each other are, as far as possible, evolutionarily related. In most existing approaches, and in this work, sequence similarity is used as a measure of evolutionary relationship, albeit an approximate one. However, more sophisticated measures are certainly possible; e.g., those that incorporate gene order (synteny).

The network alignment problem can be formulated in various ways, depending on the kind of input (pairwise vs. multiple alignments) and the scope of node mapping desired. Here, we draw an analogy from the sequence alignment problem to distinguish between local and global network alignment, the latter being the focus of this article.

Local Network Alignment (LNA): The goal in LNA is to find multiple, unrelated regions of isomorphism (i.e., same graph structure) between the input networks, each region implying a mapping independently of others. Many independent, high-scoring local alignments are usually possible between two input networks; in fact, the corresponding local alignments need not even be mutually consistent (i.e., a protein might be mapped differently under each alignment). The motivations behind local sequence alignment and local network alignment are similar— the former is often used to search for a conserved motif in the target species; the latter would be used to search for a known functional component (e.g., pathways, complexes, etc.) in a new species.

Global Network Alignment (GNA): The aim in GNA is to find the best overall alignment between the input networks. The mapping in a GNA should cover all of the input nodes: Each node in an input network is either matched to one or more nodes in the other network(s) or explicitly marked as a gap node (i.e., with no match in another network). In contrast, a LNA algorithm is essentially intended for finding similar motifs/patterns between two networks, and the mappings corresponding to different motifs may be mutually inconsistent. In GNA, however, our goal is to find a single consistent mapping covering all nodes across all input graphs. Furthermore, it must be transitive: If a_1 in G_1 is mapped to a_2 in G_2 and a_2 is mapped to nodes a_3, a_3' in G_3, then a_1 should also be mapped to a_3, a_3'. The global scope of

GNA enables species-level comparisons. Analogous to global sequence alignment, which is often used for comparing genomic sequences to understand variations between species [55], GNA may be used to compare interactomes and for understanding cross-species variations. Also, the GNA problem is related to the detection of functional orthologs, as we discuss in Results.

The focus of this chapter is on the global network alignment problem, which has previously received little attention in the literature. One can imagine using LNA to estimate GNA: Use LNA methods to compute possible matches for each protein; then select the mapping best supported overall by the alignment results. A similar approach has been used for functional ortholog detection [6]. Unfortunately, this approach is somewhat complex, and more importantly, ignores inconsistencies across local alignments so that the node matches in the final alignment might not even be mutually consistent. Instead, we propose a simpler, yet powerful algorithm.[1].

5.1 Problem Formulation

The input to the algorithm consists of two or more PPI networks $G_1, G_2, \ldots, G_k$. Each edge e may have an associated edge weight $w(e)$ ($0 < w(e) \leq 1$). In addition, other measures of similarity between the nodes may be available. In this paper, we use BLAST similarity scores, but additional measures (e.g., synteny-based scoring, functional similarity) can be incorporated.

The desired output, given only PPI network data, is the maximum common subgraph (MCS) across the graphs (i.e., the largest graph that is isomorphic to a subgraph of each graph $G_1, \ldots, G_k$) and the corresponding node-mapping. Even for the simplest case of pairwise networks (i.e., $k = 2$), MCS is known to be a NP-hard problem [38]. Thus, approximate solutions, especially for the large-sized PPI networks, are essential. Furthermore, when incorporating sequence data, the global alignment problem is no longer a pure MCS problem. To address these issues, we formulate an eigenvalue problem that approximates the desired objective.

[1] We note that in some previous works on network alignment, the distinction between "global" and "local" network alignment has centered on the relative input sizes for each. There, the term "global network alignment" is used when the input consists of roughly equal-sized networks (e.g., two species-wide networks) while "local network alignment" is used when one input is a small query network and the other is a large species-wide network. In both instances, however, the output consists of multiple local subgraphs (and corresponding local alignments). As such, we believe that both these instances are best characterized as local network alignments, regardless of input sizes.

5.2 IsoRank Algorithm

To start with, we consider the simple case of pairwise GNA. Here, the input consists of two PPI networks G_1 and G_2 (recall that the nodes of these networks correspond to proteins). Each edge e may have an associated edge weight $w(e)$ $(0 = w(e) = 1)$.

Furthermore, the input also consists of a similarity measure between the nodes of the two networks (here we use BLAST similarity scores). These scores may be defined only for some node-pairs (i.e., protein-pairs). The desired output is a mapping between the nodes of the two networks that maximizes a convex combination the following objective functions: (1) the size of the common graph implied by the mapping, and (2) the aggregate sequence similarity between nodes mapped to each other. Given the inputs, we construct an eigenvalue problem whose solution leads to a mapping between the nodes. From this mapping, the set of conserved edges can be easily computed.

Our algorithm works in two stages. It first associates a functional similarity score with each possible match between nodes of the two networks. Let R_{ij} be this score for the protein pair (i, j) where i is from network G_1 and j is from network G_2. Given network and sequence data, we construct an eigenvalue problem and solve it to compute R (the vector of all R_{ij}). The eigenvalue problem explicitly models the tradeoff between the twin objectives of high network overlap and high sequence similarity between mapped node-pairs. The second stage constructs the mapping for the GNA by extracting a set of high-scoring, mutually consistent matches from R.

Computing R (setting up the constraints): To compute R_{ij} we pursue the intuition that (i, j) is a good match if i and j's respective neighbors also match well with each other. More precisely, we require the following equality to hold for all possible pairs (i, j):

$$R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{1}{|N(u)||N(v)|} R_{uv} \quad i \in V_1,\ j \in V_2 \tag{5.1}$$

where $N(a)$ is the set of neighbors of node a; $|N(a)|$ is the size of this set; and V_1 and V_2 are the sets of nodes in networks G_1 and G_2, respectively.

These equations require that the score R_{ij} for any match (i, j) be equal to the total support provided to it by each of the $|N(i)||N(j)|$ possible matches between the neighbors of i and j. In return, each match (u, v) must distribute back its entire score R_{uv} equally among the $|N(u)||N(v)|$ possible matches between its neighbors. We note that these equations also capture non-local influences on R_{ij}: the score R_{ij} depends

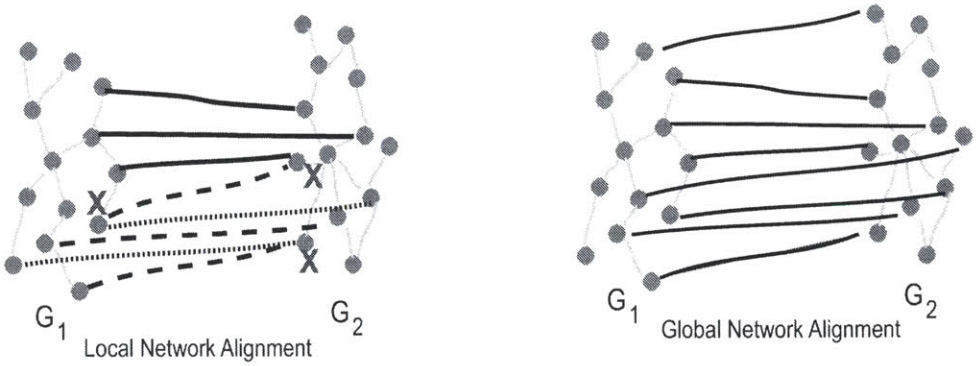


Figure 5-1: **Cartoon comparing global and local network alignments**: The local network alignment between G_1 and G_2 specifies three different alignments; the mappings for each are marked by a different kind of line (solid, dashed, dotted). Each alignment describes a small common subgraph. Local alignments need not be consistent in their mapping— the points marked with 'X' each have ambiguous/inconsistent mappings under different alignments. In global network alignment, the maximum common subgraph is desired and it is required that the mapping for a node be unambiguous. In both cases, there are 'gap' nodes for which no mappings could be predicted (here, the nodes with no incident black edges are such nodes).

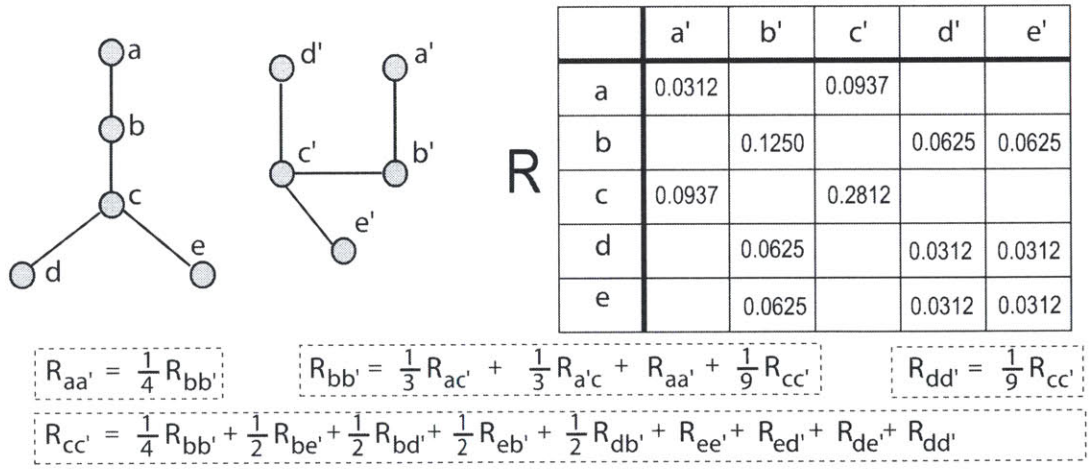


Figure 5-2: **Intuition behind the algorithm**: Here we show, for a pair of small, isomorphic graphs how the vector of pairwise scores (R) is computed. For each possible pairing (i, j) between nodes of the two graphs, we compute the score R_{ij}. The scores are constrained to depend on the scores from the neighborhood as described by Eqn. 5.1. Only a partial set of constraints is shown here. The scores R_{ij} are computed by starting with random values for R_{ij} and using the methods described below to find values that satisfy these constraints; here we show the vector R reshaped as a table for ease of viewing (empty cells indicate a value of zero). The second stage of our algorithm uses R to extract likely matches. One strategy could: choose the highest-scoring pair, output it, remove the corresponding row and column from the table, and repeat. This strategy will return the correct mapping: $\{(c, c'), (b, b'), (a, a'), (d, d'), (e, e')\}$. The $\{d, e\} \rightarrow \{d', e'\}$ mapping is ambiguous; using sequence information, such ambiguities can be resolved.

on the score of neighbors of i and j and the latter, in turn, depend on the neighbors of the neighbors and so on. The extension to the weighted-graph case is intuitive: the support offered to neighbors is now in proportion to the edge weights:

$$R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{w(i,u)w(j,v)}{\sum_{r \in N(u)} w(r,u) \sum_{q \in N(v)} w(q,v)} R_{uv} \quad i \in V_1,\ j \in V_2 \tag{5.2}$$

Clearly, Eqn. 5.1 is a special case of Eqn. 5.2 when all the edge weights are 1. We can rewrite Eqn. 5.1 in matrix form (Eqn. 5.2 can be similarly rewritten):

$$\begin{aligned} R &= AR \\ A[i,j][u,v] &= \begin{cases} \frac{1}{|N(u)||N(v)|} & \text{if } (i,u) \in E_1 \text{ and } (j,v) \in E_2 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{5.3}$$

where A is a $|V_1||V_2| \times |V_1||V_2|$ matrix and $A[i,j][u,v]$ refers to the entry at the row (i,j) and column (u,v) (the row and column are doubly-indexed).

Another interpretation of the above equations is that they describe a random walk on the product graph of $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. We define $G^* = (V^*, E^*)$ where $V^* = V_1 \times V_2$ and $E^* = \{(\,(i,j),(u,v)\,) \mid (i,u) \in E_1,\ (j,v) \in E_2\}$. Also, if G_1 and G_2 are weighted, so is G^*: $w(\,(i,j),(u,v)\,) = w(i,u)w(j,v)$. We now specify a random walk among the nodes of G^*: from any node we can move to one of its neighbors, with a probability proportional to the edge weight:

$$P(s_t = (i,j) \mid s_{t-1} = (u,v)) = \frac{w(i,u)w(j,v)}{\sum_{r \in N(u)} w(r,u) \sum_{q \in N(v)} w(q,v)} \tag{5.4}$$

where s_t is the node occupied at time t. Eqns. 5.1, 5.2 and 5.3 can now be interpreted as defining R to be the stationary distribution of this random walk (its transition matrix is A). Thus, a high R_{ij} implies that the node (i,j) of G^* has a high probability of being occupied in the stationary distribution.

The vector R is determined by finding a non-trivial solution to these equations (a trivial solution is to set all R_{ij}s to zero). As Eqn 5.3 indicates, R is the principal eigenvector of A.

In Fig 5.2, we illustrate, on a pair of small graphs, how the equations capture the graph topology; their solution also confirms our intuition: node pairs that match well have higher R_{ij} scores.

Computing R (solving the constraints): In general, to solve the above equations, we observe that these equations describe an eigenvalue problem (see Eqn. 5.3). The value of R we are interested in is the

principal eigenvector of A. Note that A is a stochastic matrix (i.e., each of its columns sums to 1) so that the principal eigenvalue is 1. Also, for numerical stability purposes we require that R be normalized, i.e., $|R|_1 = 1$. In the case of biological networks, A is typically a very large matrix (about $10^8 \times 10^8$ for fly-vs.-yeast GNA); however, A and R are both very sparse, so R can be efficiently computed by iterative techniques. We use the *power method*, an iterative technique often used for large eigenvalue problems. The power method repeatedly updates R as per the update rule: $R(k+1) \leftarrow AR(k)/|AR(k)|$, where $R(k)$ is the value of the vector R in the k-th iteration and has unit norm. In case of a stochastic matrix (like A), the power method will provably converge to the principal eigenvector; the convergence can be sped up significantly by a judicious choice of the initial value $R(0)$. As we describe shortly, a good initial value $R(0)$ is often available in our case.

The incorporation of other information, e.g. BLAST scores, into this model is straightforward. Let B_{ij} denote the score between i and j; for instance, B_{ij} can be the Bit-Score of the BLAST alignment between sequences i and j. B_{ij}s need not even be numeric— they can be binary. Let B be the vector of B_{ij}s. We first normalize B: $E = B/|B|$. The linear system of equations is then modified to

$$R = \alpha AR + (1-\alpha)E \quad \text{where} \quad 0 \leq \alpha \leq 1. \tag{5.5}$$

Eqn. 5.5 is solved by similar techniques as Eqn. 5.3.

Also, node matches based purely on sequence similarity are an approximation to the node mappings desired; hence, the vector E is a good choice for the initial value $R(0)$ in the power method. We emphasize that this choice of starting value does not change the final value of R— it just speeds up the computation. We emphasize that in each iteration the vector E is used.

In this computation, α controls the weight of the network data (relative to sequence data), e.g., $\alpha = 0$ implies no network data will be used, while $\alpha = 1$ indicates only network data will be used. Tuning α allows us to analyze the relative importance of PPI data in finding the optimal alignment. Until now, such an analysis has been difficult to perform, even for existing local network alignment methods.

Compute R for multiple species: When performing multi-species GNA, the eigenvalue computation described above is performed independently for each pair of networks. The resulting R_{ij} values are concatenated into the vector R. For each node pair (s,t), the score R_{st} is present if and only if s and t are nodes in different networks. Once R has been computed, we extract the node mappings from it.

Extracting the mapping from R (Simple Case): Suppose that there are only two networks that need to

be aligned to each other. Furthermore, suppose we restrict the allowed node-mappings to those where, for each node in a species, there is at most one corresponding node in the other species. The motivation behind imposing this restriction is to simplify the problem while still retaining biological relevance. The at-most-one-correspondence requirement has an intuitive interpretation: the corresponding nodes are then the closest functional orthologs of each other in the two species.

For this special case, there is an intuitive solution. The vector R can be interpreted as describing a bipartite graph between the nodes of the two species: an edge (i, j) exists in this graph if and only if the score $R_{ij} > 0$. Furthermore, the weight of the edge is R_{ij}. The set of allowed mappings as per the restriction described in the previous paragraph then corresponds to the set of possible matchings in this bipartite graph. Furthermore, our goal is to extract the set of mutually-consistent, pairwise matches (p, q) such that the sum of their scores R_{pq} is maximized. This is precisely the maximum-weight bipartite matching problem, a problem with a well-known polynomial time solution [70]. We compute the maximum-weight matching in this bipartite graph and output the paired nodes. Any remaining unpaired nodes are designated as gap nodes. This algorithm guarantees the set of matches that satisfy our criterion.

While this algorithm does give good results, in practice we found that the following greedy matching algorithm sometimes performs even better from a biological perspective: identify the highest score R_{pq} and output the pairing (p, q). Then, remove all scores $R_{p\cdot}$ and $R_{\cdot q}$ involving p or q. We then repeat this process until the list is empty. In the bipartite graph, this strategy corresponds to removing, at each step, the maximum weight edge and the incident nodes.

Extracting the mapping from R (General Case): The more general case is when a node can be mapped to more than one node in another species. The mapping produced here is of the same form as Clusters of Orthologous Genes (COGs) [86]: The entire set of nodes across all networks is partitioned, each partition corresponding to a set of nodes mapped to each other. Each set may contain zero, one or many nodes from each species. The intuition here is that the proteins in a single set are functional orthologs of each other, i.e., are evolutionarily related and perform the same function in their respective species.

To construct such a partition of genes from the set of scores R computed in the previous approach, we design an algorithm that searches for sets of genes such that each set obeys the following requirements: (1) each gene in the set has high pairwise R scores with most other genes in the set; (2) there are no genes outside each set with this property; and, (3) there are a limited number of genes from each species. This limit varies from species to species: more genes from *H. sapiens* are allowed in the set than from *S.*

cerevisiae, reflecting the intuition that there is greater gene duplication in the former.

Our algorithm computes each set of orthologous proteins by identifying a seed pair of match nodes and extending it by using a modified greedy algorithm. We first construct a k-partite graph H from the scores R. Each of its k parts contains nodes from one of the input networks. Edges are only allowed between nodes from different parts. The presence of an edge e_{ij} implies that node i (from G_1) can potentially be mapped to j (from G_2), i.e., $R_{ij} > 0$; the edge-weight R_{ij} indicates the strength of the potential match.

While the k-partite graph H has any edges remaining:

1. Select the edge e_{ij} with the highest score (let i be from G_1 and j from G_2). Initialize a new match-set with i and j as its initial members.

2. In every other species $G_3, \ldots, G_k$, if a node l exists such that (i) R_{il} and R_{jl} are the highest scores between l and any node in G_1 and G_2, respectively and, (ii) the scores $R_{il} \geq \beta_1 R_{ij}$, and $R_{jl} \geq \beta_1 R_{ij}$, add it to the set. This set of nodes forms the primary match-set; it has at most one node from each species.

3. Add upto $r - 1$ nodes from different parts of the graph to the primary match-set. Suppose u (from G_x) is in the primary match-set. Then, a node v (from G_x) is added to the set if $R_{vw} \geq \beta_2 R_{uw}$ for each node $w (w \neq u)$ in the primary set.

4. Remove from H all of the nodes in this match-set and their edges.

Here, the parameters r, β_1, β_2 are user-defined ($0 < \beta_1, \beta_2 < 1$); we chose their values such that the functional coherence (defined in next section) of the resulting sets of matched nodes was maximized.

Once a comprehensive alignment has been computed, the corresponding subgraph in the GNA can be identified relatively easily. For example, if a_1 is aligned to a_2, and b_1 is aligned to b_2, the output subgraph should contain an edge between (a_1, a_2) and (b_1, b_2) if and only if both the input networks contain supporting edges (i.e., (a_1, b_1) in G_1 and (a_2, b_2) in G_2). When edges also have associated weights, formalizing the intuition depends on how the edge weights are being interpreted; for example, we could require that the combined weight be higher than a threshold or that the minimum of the two be greater than a threshold.

5.3 IsoRank-N: Using Spectral Partitioning

In this section, we discuss an alternative approach to extracting a mapping from the set of scores R, by posing the problem as a clique-finding task. If the data were noise-free and complete, each set of functional orthologs would be a clique in the graph H (defined above) described by the scores R. Furthermore, finding the optimal mapping would essentially be the max-weight clique identification problem. While this problem itself is NP-hard, an added complication is that the PPI data is noisy and incomplete. Here, we need to find near-cliques of large weights. We take a graph-partitioning approach to solving this problem.

We start with an ordering of the PPI network networks $G_1, \ldots, G_k$ (later, we discuss which orderings work better than others). Starting with the first network, for every protein v in the chosen network, we construct the *star* subset S_v of nodes which are connected to v with a large weight, i.e., $R_{uv} \geq \gamma_1$ for all $u \in S_v$. Intuitively, each such set is the superset of one (or more) cliques in H. We order the sets S_v in decreasing order of total weight $w_{tot} = \sum_{a,b \in S} R_{ab}$. For each successive set S_v, we compute the subgraph of H implied by its members and identify a high-weight clique-like neighborhood in this subgraph. The nodes in this near-clique correspond to a set of a functional orthologs. We remove these nodes from further consideration and proceed down the list of subsets S_v, until no nodes are left to be matched.

The problem then reduces to identifying high-weight near-cliques in the subgraph implied by nodes in S_v. We find an approximate solution for this using spectral approaches. Instead of finding a maximally weighted clique containing v, we find a low-conductance set containing v. The *conductance*, $\Phi(S)$, of a subset S of a graph H is a measure of the separation between S and $H \setminus S$ (i.e., the subgraph of H formed by nodes not in S). It is the ratio of the edge-cut that separates S (from $H \setminus S$) to the maximum edge-weight in either S or $H \setminus S$. More formally, $\Phi(S) = \frac{\sigma(S)}{\min\{vol(S), vol(H \setminus S)\}}$, where $\sigma(S) = |\{(v_x, v_y); v_x \in S, v_x \in H \setminus S\}|$ and $vol(S) = \sum_{i \in S} deg(v_i)$. This measures provides a very natural measure of the extent to which the nodes in S are co-clustered, relative to the other nodes in H.

Anderson et al. [2] showed that a low-conductance set containing v can be computed efficiently via the personalized PageRank vector of v. A personalized PageRank vector $Pr(\gamma, v)$ is the stationary distribution of the lazy random walk on S_v in which at every step, with probability γ, the walk jumps back to v and with probability $1 - \gamma$ performs a lazy random walk with transition probabilities proportional to the values in R (the $lazy$ part means that with probability 0.5, the walk does not move). The desired PageRank vector $Pr(\gamma, v)$ can be found by solving the following equation:

$$Pr(\gamma, v) = \gamma\chi_v + (1-\gamma)Pr(\gamma, v)W, \tag{5.6}$$

where $0 \leq \gamma < 1$, and $\chi_v(x) = \delta_{x,v}$ is the indicator vector of v, $W = \frac{1}{2}(I + D^{-1}R)$ is the lazy random walk transition matrix and D is the diagonal of the column sums of R.

While highly efficient, the above "star" method has the limitation it only contains one node (v) in $S*_v$ from the original network G_i. To address this, we merge the "stars" as follows: given two stars $S*_{v_1}$ and $S*_{v_2}$, where v_1, v_2 are in the same PPI network G_i, we combine the two subgraphs if every neighbor of v_1 in $S*_{v_1}$ is connected to v_2 and vice versa. We can now describe the full algorithm

The IsoRankN Algorithm

Given k PPI networks $G_1, G_2, \ldots, G_k$ and a threshold β, IsoRankN proceeds as follows:

1. Using the methods described in the previous section, compute the scores R_{ij} and the corresponding k-partite graph H
2. For every node v in H, compute the star $S_v = \{v_j \in N(v) | w(v, v_j) \geq \beta \max_j w(v, v_j)\}$, where $N(v)$ is the neighborhood of v in the graph H
3. Pick an arbitrary remaining PPI network G_l and order the proteins $v \in G_l$ by the sum of edge weights in the induced graph on S_v. Spectrally partition S_v to obtain $S*_v$, after excluding proteins already assigned to clusters.
4. Merge every pair of clusters $S*_{v_1}$ and $S*_{v_2}$ in which $\forall v_i \in S*_{v_2} \backslash \{v_1\}, w(v_1, v_i) \geq \beta \max_j w(v_1, v_j)$
5. Repeat steps 3 and 4 until all proteins are assigned to a cluster.

5.4 Results: Two-Species Case

In this section, we describe the results of two-way global alignment of the *S. cerevisiae* and *D. melanogaster* PPI networks, the two species with the most available network data. We also evaluate the algorithm's robustness to error, the sensitivity to the parameter α, and discuss heuristics for choosing an appropriate α.

The PPI network data for the species was retrieved from the BIOGRID [83], DIP [92], and HPRD [67] databases, and the sequence data was retrieved from Ensembl [12]. The edges in the PPI networks did not have associated weights. We applied IsoRank to this pair of networks, using it to identify the common subgraph.

The common subgraph corresponding to the global alignment between the yeast and fly PPI networks has 1420 edges (where $\alpha = 0.6$). While the amount of overlap might seem relatively small (both the networks have more than 25000 edges each), it is not surprising. The is primarily because the currently available PPI datasets are noisy and incomplete. They are known to contain many false-positives. Also, current PPI data is far from comprehensive; e.g., the fly network has no known PPIs for about 6500 proteins (almost 50% of the genome). As these issues get resolved, we expect the size of the global alignment to grow substantially. Nevertheless, the current global alignment already provides many valuable insights.

Such global alignment of PPI networks provides insights typically unavailable from local network alignment (LNA) approaches like Pathblast [54]. The common subgraph described above has many disconnected components, an artifact we believe is related to the noise and completeness issues with the data. Still, its largest component which has 35 edges (Fig. 5-3) is significantly larger than any common subgraph we could identify using Pathblast. The longest pathway-like component identified by the latter had 4 nodes, and the largest complex-like component had 16 nodes. Furthermore, some of the LNA methods are limited [54, 58] in that they are well-suited to identifying only certain specific topologies (e.g. linear pathways or clique-like protein complexes). In contrast, the components of the global alignment span various topologies, from linear pathways (Fig. 5-4(a)) to components corresponding to protein complexes (Fig 5-4(d)). We emphasize that our components were discovered simultaneously— they are just subgraphs of the larger alignment graph. Many of our discovered components are de-facto *functional modules* (though not in the sense Flannick *et al.* [33] use the term): they are enriched in proteins involved in a single biological process and can thus be mapped to specific cellular functions (e.g., see Fig 5-4(d)). These functions range from various signaling cascades (Fig. 5-4(b)) to core cellular functions like ribosomal synthesis and function (Fig. 5-4(c)), DNA transcription and translation, cell division etc. The preponderance of core cellular functions in the conserved subgraph is not too surprising— it is exactly these mechanisms that are likely to be highly conserved across species.

The global alignment may be used to predict protein function. For example, Fig 5-4(d) shows a subgraph of the global alignment, most of the proteins in which are involved in SCF ubiquitin ligase activity. Hence, we predict the function of two hitherto-unannotated fly proteins CG7148 and CG13213 as being

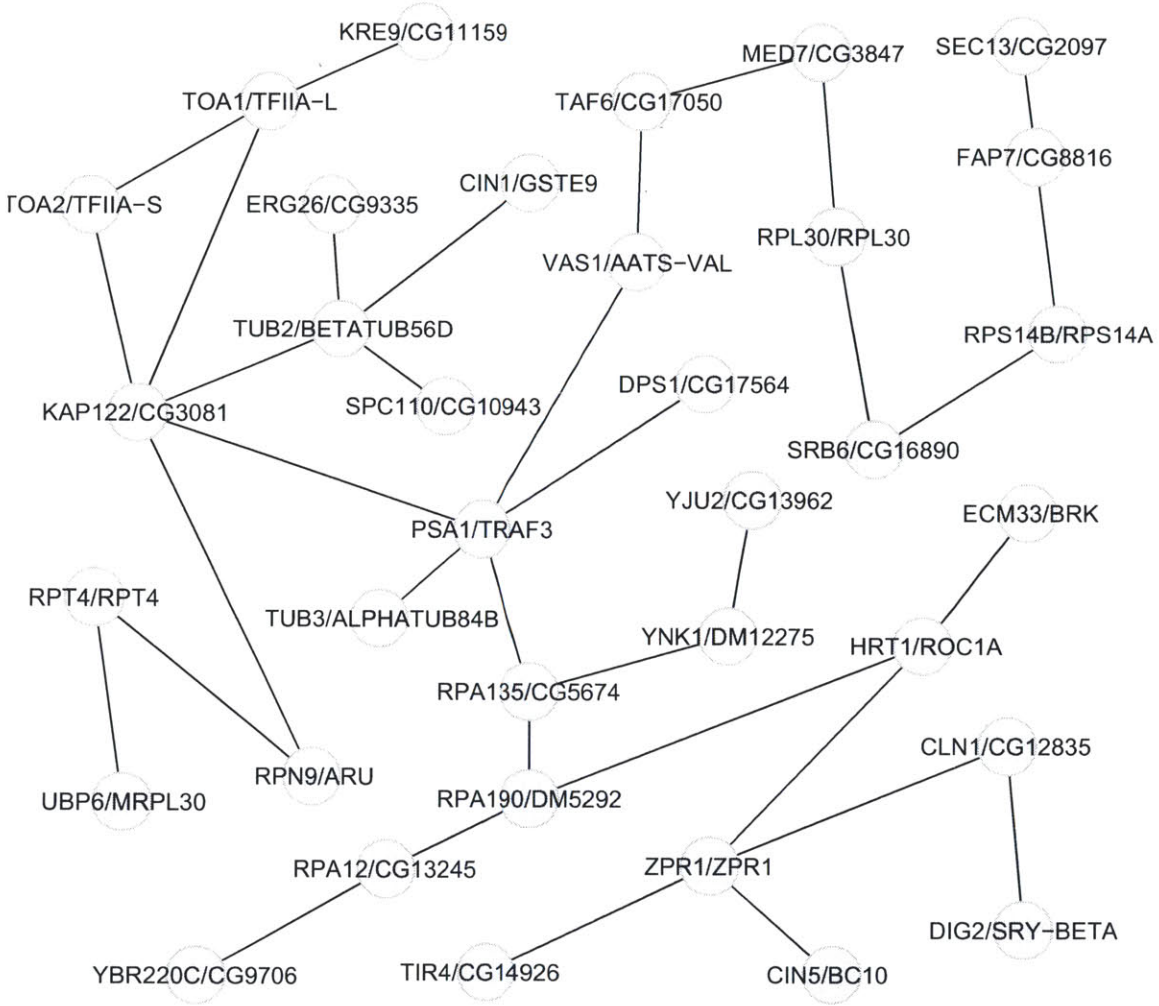
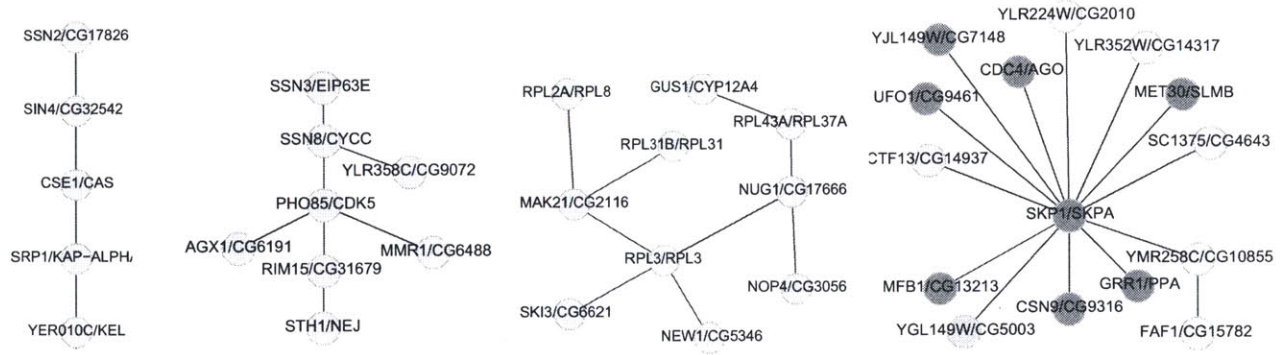


Figure 5-3: **Largest connected component of the yeast-fly Global Network Alignment:** The node labels indicate the corresponding "yeast/fly" proteins (the two separated by a "/"). The proteins in this graph span a variety of functions: metabolic, signaling, transcription etc. For a discussion of this subgraph's size, see text.



(a) Pathway (b) Kinases (c) Ribosome Complex (d) Ubiquitin Ligase

Figure 5-4: **Selected subgraphs of the yeast-fly GNA:** The node labels indicate the corresponding "yeast-/fly" proteins (the two separated by a "/"). The subgraphs span a variety of topologies and are often enriched in specific functions (c) and (d). In (d), the nodes for which at least one of the corresponding proteins is known to be involved in ubiquitin ligase activity are shaded.

involved in ubiquitin protein ligase activity. In support of this, we note that the FlyBase database [20] indicates that the involvement of these proteins in ubiquitin ligase activity has been postulated before in the literature. Of course, more sophisticated methods to transfer annotation may perform even better at elucidating function of such proteins [68].

Evaluating the algorithm's error tolerance: Our simulations indicate that the algorithm is tolerant to error in the input (Fig 5.4); this is valuable since PPI networks have high false positive and false negative rates. To evaluate the algorithm's error-tolerance, we first extracted a 200-node subgraph of the yeast PPI network. We then randomized a fraction p of its edges using the Maslov-Sneppen trick that preserves node degrees [65]: we randomly choose two edges (a, b) and (c, d), remove them, and introduce new edges (a, d) and (c, b). We then computed a GNA between these two graphs, with $\alpha = 1$ and $\alpha = 1 - 10^{-6}$. For each choice of p, we created 5 such randomized graphs and computed the average fraction of nodes that are mapped to themselves in the original graph after a GNA.

Using $\alpha = 1$ results in a significant underestimate because there often are multiple possible isomorphism-preserving mappings between two isomorphic graphs (e.g., see Fig 5.2) and our algorithm— even if working correctly— might choose a mapping that does not preserve node labels. Adding a very small amount of sequence information ($\alpha = 1 - 10^{-6}$) helps avoid this, but also results in a slight overestimate. We believe the true curve (for Fig 5.4) is closer to the top curve than the bottom one. Clearly, the algorithm makes very few mistakes when the error rate p is low and even for fairly high error rates (20-50%), its

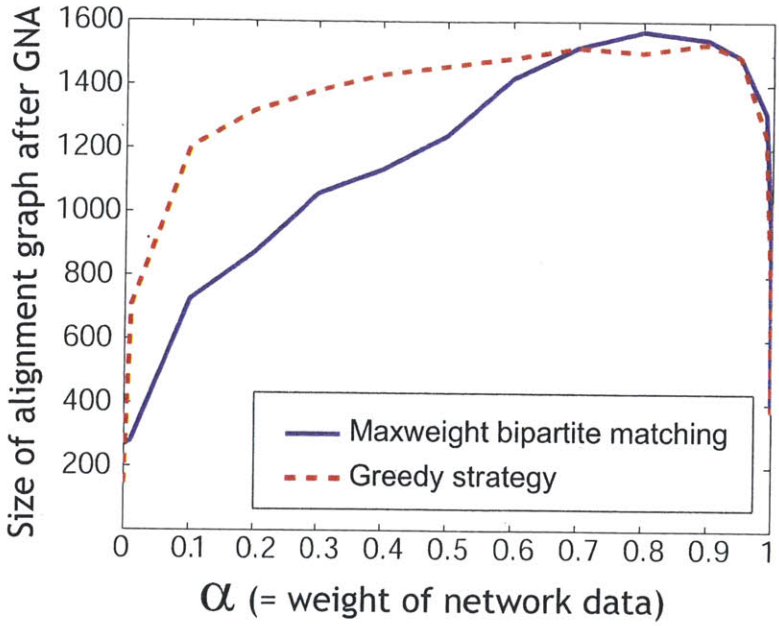


Figure 5-5: **Impact of α on the size of the alignment graph.**

performance degrades smoothly and very slowly. When computing the yeast-fly GNA, we assigned a significant weight to sequence information ($\alpha = 0.6$).

Evaluating the influence of α: As α increases, so does the importance of network data in the alignment process, for both the greedy strategy and the maximum weight bipartite matching strategy (Fig 5.4). In line with our expectations, the size of the common subgraph depends on this parameter: $\alpha = 0$ results in a graph with 266 edges, while $\alpha = 0.9$ results in 1544 edges (for the greedy strategy). Intriguingly, as α gets very close to 1, the common graph's size *decreases*. We believe that this discrepancy is an artifact of the current PPI data sets being noisy and covering the interactome only partially, resulting in a relatively small overlap between the yeast and fly PPI networks. Consequently, in absence of any other information a random mapping of nodes between the two networks might satisfy Eqn.5.1 better than the one corresponding to the "true" alignment. The use of sequence-based scores helps mitigate this, by directing the algorithm towards the true alignment.

When choosing the most appropriate value of the free parameter α, we rejected the choice corresponding to the largest common subgraph size— the input networks are noisy and conserved edges may be simply due to noise; thus, the α leading to the largest-size subgraph may not be a biologically appropriate choice. Instead, for each choice of α, we compared the resulting node mappings to sequence-based ortholog predictions from the Inparanoid database [77] and chose the α (= 0.6) that resulted in the greatest

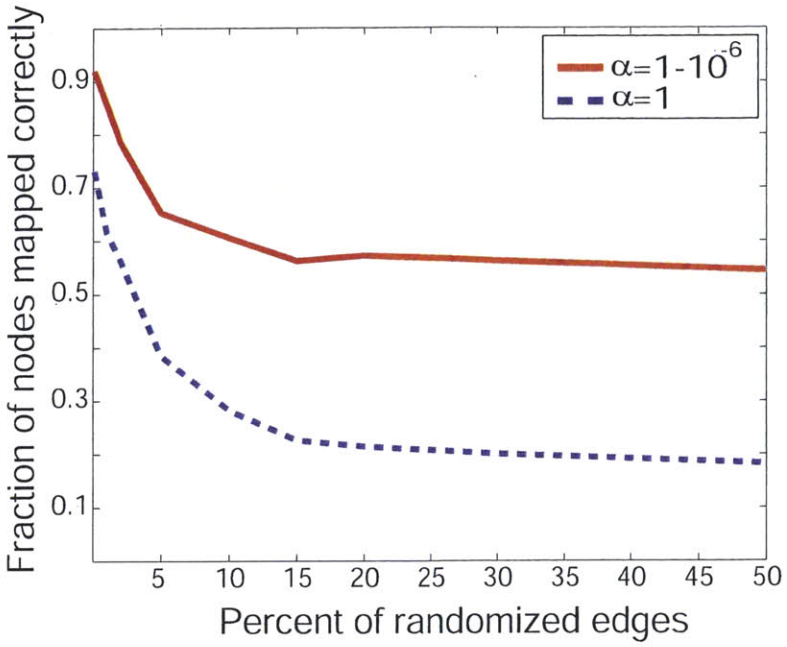


Figure 5-6: **Effect of PPI errors on the algorithm's performance** We believe the solid (red) curve slightly overestimates the algorithm's performance, while the dashed (blue) curve grossly underestimates it (see text).

overlap with these. While this approach is conservative and might undervalue the network component during the alignment, it also lowers the adverse impact of noise in the PPI data.

The differences between the node pairings found by our algorithm and those from Inparanoid broadly fall into two categories: (1) those corresponding to low R_{ij} values indicating low confidence of our approach in that mapping, and (2) functional orthologs where the use of network data genuinely changes the node mapping. We discuss the latter in more detail later in this section.

Comparing global and local alignment results: Our global alignment results compare favorably to the those of NetworkBlast [1] (an implementation of PathBlast) and sequence-only approaches. We compared the aggregate set of local alignments from NetworkBlast with our global alignment. Each local alignment defines one-to-one matches between some yeast and fly proteins. Many of the matches from our global alignment are seen in these local alignments: of the 701 matched protein-pairs in the former that consist of proteins seen in at least one local alignment, 83% (582) of the pairs are also observed in one or more local alignments. However, there are many overlapping local alignments, resulting in ambiguity and inconsistency: averaged across the entire set of local alignments, a yeast protein is aligned to 5.36 different

fly proteins. Sometimes, such ambiguity may be biologically meaningful, e.g., in instances of gene duplication. However, the degree of ambiguity in some of the PathBlast results is clearly implausible. For example, the yeast protein SNF1, a Serine-Threonine Kinase (STK), is matched to 71 different fly proteins. In fact, PathBlast results for many of the yeast STKs are very ambiguous– over the set of 72 yeast proteins annotated as STKs, the average number of matching fly proteins per yeast STK is 29.3. STKs are part of many important signaling pathways, e.g, the MAPK, JNK and AKT cascades. Sequence-only approaches. (e.g. Inparanoid) too have performed poorly at ascertaining the correspondence between yeast and fly STKs: Inparanoid does not predict any fly orthologs for 58 of the 72 yeast STKs. Thus the use of GNA to resolve this ambiguity in correspondence is particularly valuable.

GNA and functional orthologs: In analogy with sequence-based comparative genomics methods [55], we apply IsoRank to the detection of functional orthologs (i.e., sets of proteins that perform the same function in two or more species) by exploiting the strong connection between these two problems: proteins that are aligned together in the global alignment should have similar interaction patterns in their respective species and are thus likely to be functional orthologs. There has been a lot of recent interest in the discovery of functional orthologs (FO). In particular, Bandyopadhyay *et al.* [6] took a fairly complex approach to FO detection between yeast and fly through local network alignment (LNA): first, possible FOs for a protein are short-listed using a sequence-only approach; then, using a probabilistic technique (based on Markov Random Fields) and the results of a LNA of the yeast and fly networks (performed using PathBlast), the probability of each short-listed pair of proteins being true FOs is computed.

The results of IsoRank compare favorably with Bandyopadhyay *et al.*'s. Our method has the advantage that it guarantees the predicted sets of FOs will be mutually consistent and achieves higher genome coverage— PathBlast's yeast-vs.-fly local alignments cover only 20.56% of the genes covered by our global alignment. In many cases the FO predictions between the two methods are partially or fully consistent (see Table 1), i.e, FOs predicted by our method are also the likely FOs predicted by their method. Furthermore, their method often proposes multiple FOs for a protein, and our method resolves the ambiguity in their results. In a few other cases, predictions of the two methods differ. At least in some such cases, our method's predictions are better supported by evidence. For example, our method predicts *Bic* (in fly) as the FO of *Egd* (in yeast). Bandyopadhyay *et al.*'s method is ambiguous here as *Bcd*, its predicted FO of *Egd*, is also predicted as a FO of *Btt1*. Furthermore, there is experimental evidence that both *Egd* and *Bic* are components of the Nascent Polypeptide-Associated Complex (NAC) in their respective species, lending support to our prediction; in contrast, *Bcd* does not seem to be involved in NAC.

Protein (species)	Predicted Functional Ortholog by Our Method	Related Predictions from (Bandyopadhyay et al.)		Remarks
		Yeast/Fly pair	Prob.	
Gid8 (yeast)	CG6617	Gid8/CG6617 Gid8/CG18467	76.51% -	Our predictions consistent with Bandyopadhyay et al.[1]
Tpm2 (yeast)	Tm1	Tpm2/Tm1	-	Consistent predictions.[1]
Tpm1 (yeast)	Tm2	Tpm1/Tm2	43.98%	Consistent predictions.[1]
Gpa1 (yeast)	G-oα47a	Gpa1/G-oα47a Gpa1/G-ia65a	41.53% -	Consistent predictions.[1]
Rpl12 (fly)	Rpl12a	Rpl12a/Rpl12 Rpl12b/Rpl12	48.39% -	Consistent predictions.[1]
Btt1 (yeast)	CG11835	Btt1/CG11835 Btt1/Bcd	70.5% 40.86%	Consistent predictions.[1]
CG18617 (fly)	Vph1	Vph1/CG18617 Stv1/CG18617	43.53% 38.44%	Consistent predictions.[1]
Kap104 (fly)	Trn	Kap104/Trn Kap104/CG8219	40.64% 46.78%	Partially consistent predictions.[2]
Act1 (yeast)	Act5c	Act1/Act5c Act1/Act42a Act1/Act87e Act1/Act88f Act/CG10067	39.56% 39.24% 43.53% 40.17% 38.20%	Partially consistent predictions.[2]
Kel2 (yeast)	CG12081	Kel2/CG12081 Kel1/CG12081	- 45.41%	Partially consistent predictions.[2]
Cmd1 (yeast)	Cam	Cmd1/Cam Cmd1/And	35.90% 44.39%	Partially consistent predictions.[2]
Hsc70-4 (fly)	Ssa3	Hsc70-4/Ssa3	-	Partially consistent predictions.[2]

Table 5.1: **Interpreting two-way global alignment results as functional orthologs (FOs)**: Comparison of our results with Bandyopadhyay *et al.*'s results [6]. Our method is often consistent with their results and, moreover, often resolves the ambiguity in their predictions. [1]Our predicted FO for the protein matches Bandyopadhyay *et al.*'s predicted FO, or the most likely FO if their method predicted multiple FOs.[2]Our predicted FO for the protein is one of the likely FOs predicted by Bandyopadhyay *et al.* (but not the most likely one).

	IsoRankN	IsoRank	Graemlin$_{1K}$ [33]	Graemlin$_{2K}$ [33]	NetworkBlast [1]
Mean entropy	0.274	0.685	0.857	0.552	0.907
Mean normalized entropy	0.179	0.359	0.451	0.357	0.554
Exact cluster ratio	0.380	0.253	0.306	0.355	0.291

Table 5.2: **Consistency of IsoRank & IsoRankN's multi-species predictions**. IsoRankN and IsoRank have lower (i.e. better) GO entropy scores than the other approaches. IsoRankN also produces more ortholog-sets where all the genes have exactly the same GO annotation. The two instances of Graemlin above refer to the different training set sizes for the algorithm.

5.5 Results: Multi-Species Case

We performed a global alignment of PPI networks from 5 eukaryotic species: fly, yeast, worm, mouse and human. From this alignment, we inferred functional orthologs between the various species. To evaluate these results, we looked at the coverage and consistency of our results. Here, *coverage* refers to the set of genes for which orthology relationships could be inferred. Some network alignment methods may have low coverage, especially if they rely on the availability of functional annotation to infer the alignment. ISORANK and IsoRankN have significantly better coverage than Graemlin [33]. We measure this by counting the number of genes successfully matched to at least one gene in another species and the number of unique clusters (ortholog-sets) produced. In total, IsoRank produces 12848 clusters covering 48978 genes; in contrast, Graemlin produces 4306 clusters covering 20903 genes. The difference is starker for clusters with genes from more than two species. When considering clusters that have genes from all five species, IsoRankN produces 2056 clusters 12715 genes while Graemlin produces 58 clusters with 1467 genes.

We also measure the consistency of the functional ortholog predictions. Here, *Consistency* of the results refers to whether the functional orthologs predicted do have the same function across various species. To quantify this notion, we introduce a way to measure the functional coherence of orthology predictions:

$$H(S) = H(p_1, p_2, \ldots) = -\sum_{i=1}^{d} p_i \log p_i \tag{5.7}$$

Here, $H(S)$ measures the entropy in the distribution of GO terms for genes in the set S. Intuitively, lower $H(S)$ implies that more genes in the set share the same GO terms and, thus, the consistency is higher. As shown in Tab. 5.2, IsoRank and IsoRankN achieve significantly better consistency than other multi-species network alignment algorithms.

5.6 Conclusion

In this section, we focus on the global network alignment problem, and describe an intuitive yet powerful algorithm for computing the global alignment of two PPI networks; in contrast, much of the previous work has been focused on the local alignment problem. Our algorithm, IsoRank, simultaneously uses network and sequence information and is tolerant of noise in the inputs; furthermore, it is easy to control the relative weights of the network and sequence information in the alignment. We use IsoRank to compute a global alignment of the *S. cerevisiae* and *D. melanogaster* PPI networks. The results provide valuable insights about the conserved functional components between the two species. They also allow us to predict functional orthologs between the fly and yeast; the quality of our predictions compare favorably with previous work.

Our algorithm is similar— in spirit— to Google's PageRank algorithm, which ranks web-pages in the order of their "authoritativeness". The intuition behind the two algorithms has a similar flavor: in PageRank, a page has a high score if many pages with high scores link to it. The intuitions are also formalized similarly– by constructing an eigenvalue problem. Our actual algorithm is quite distinct from PageRank: in our case the input is a pair of undirected, weighted graphs and the output is an alignment; PageRank's input is a directed, unweighted graph (where the nodes indicate web-pages and directed edges, hypertext links), and it outputs node rankings.

Chapter 6

Influence Flow: Integration of PPI and RNAi Data

Signaling networks are some of the most interesting, challenging, and medically-relevant parts of the larger cellular system. A signaling network is a cellular subsystem that captures the pattern and sequence of interactions through which the cell receives an extracellular signal (typically, a small molecule) at its membrane, recognizes it, and initiates a sequence of protein interactions inside the cell, the final impact of these being to modulate the activity and expression of a large set of genes and proteins. These changes are manifested as a cell-level response to the received signal [56, 53]. From a medical perspective, signaling networks play a disproportionately important role in many diseases. Because of their role as cellular "switches", malfunctions in these lead to significant anomalies in cell behavior. Such malfunctions have been directly linked to many of the cancers[1], diabetes, and many genetic diseases [74, 29, 44]. Various signaling networks have been studied and their core components (and sequence of interactions) seem to be conserved across eukaryotes. Examples of such signaling networks include: the MAP Kinase network [17], the Wnt network [22] , the JAK-STAT network [72] etc. The responses brought about by the signaling mechanism are of varied types: cell growth, proliferation (i.e. cell division), differentiation (e.g. from a stem cell to a muscle cell), and even apoptosis (cell death). Clearly, understanding signaling networks is of crucial biological and medical importance.

Structurally, a signaling network can be represented as a directed network: each node corresponds to a protein/gene and each interaction is indicated by a directed edge. The edge's direction indicates

[1]Often, the cancerous cell fails to respond to extracellular signals asking it to stop proliferating

the direction of signal-flow. Furthermore, many of the edges may be annotated to indicate whether the interaction results in activation or repression of the downstream node. The interactions within a signaling network may be of various types: phosphorylation, dephosphorylation, transient interaction, complex formation, and possibly, protein-DNA interactions. The structure of signaling networks enables a great amount of flexibility when transmitting the signal: they can amplify the received signal, attenuate it, distribute it to multiple recipients, integrate multiple signals etc. [73]

In this chapter, we propose a computational technique which generates hypotheses about a specific signaling network's structure. We combine PPI networks with RNAi data specific to a particular signaling subsystem to produce hypotheses about the structure of that subsystem. The RNAi data is generated from a functional genomic screen of a specific signaling pathway. These screens work as follows: a known end-effector gene of the pathway is chosen as the reporter gene (e.g., *Erk* in the MAPK pathway). Every other gene in the genome is systematically knocked-down using RNAi and the effect on the reporter is measured. The experiment produces a list of genes (*hits*) that significantly influence the reporter and, for each hit, a score indicating the relative strength of its influence [35]. The second input to our method is genome-wide PPI data (protein-DNA interactions can also be included).

Our algorithm is based on the Occam's Razor principle. Given the inputs, we search for the simplest arrangement of nodes in a directed graph, such that (1) the nodes correspond to proteins, (2) the directed edges correspond to hypothesized interactions, and (3) the graph's topology is consistent with the input PPI and RNAi data and with the known biology of the chosen signaling subsystem. We do this by borrowing ideas from the multicommodity flow literature to construct a linear program whose solution corresponds to our desired graph. This graph can then be interpreted as a collection of high-confidence hypotheses about the topology of the signaling subsystem.

Our work is motivated by the urgent need for computational techniques to supplement experimental methods of discovering signaling topology. This need stems from an appreciation that signaling networks are significantly more complex than previously thought, and that a very large set of hypotheses regarding their structure still need to be tested [71, 36]. The classical understanding of signaling networks was that a typical network is essentially a linear pathway with less than 10 component, and with very few connections between different signaling subsystems. More recently, however, experiments like synthetic lethality, genetic interactions and RNA interference (RNAi) have demonstrated that the number of genes/proteins that influence a signaling subsystem is much larger. Furthermore, they suggest that the most common topology of such subsystems is a general network and not a linear pathway. Also, there seems to be

significant cross-talk between the various signaling networks [25]. Clearly, much of the topology of signaling networks remain to be discovered. Unfortunately, while these experimental techniques provide *some* information about the signaling networks' structure, a clear, deep understanding of this structure can only be achieved by validating specific interactions by rigorous *in-vivo* and *in-vitro* experiments, which can be time-consuming. Hence, there is a need for computational methods to identify high-likelihood hypotheses which can be tested first.

Despite its importance, the computation discovery and prediction of signaling networks has remained a challenging problem. One key reason for its difficulty is the lack of appropriate high-quality data. Yeang *et al.* attempted to predict signaling networks starting from first principles, by combining PPI, protein-DNA and gene knockout data [95, 96]. Their method is very sophisticated and demonstrated significant promise; however, quality and coverage issues with the data limited its biological usefulness. In another approach, Sachs *et al.* used high-quality single-cell data specifically generated for their analysis [78]. In their experiment, a limited set (at most 12) of proteins were tagged and their phosphorylation levels were measured under different conditions, at a single-cell resolution. Using a Bayesian approach, Sachs *et al.* were able to construct a signaling network of the selected proteins; their predicted network had striking agreements with the biologically-determined structure. However, the experiment they relied is difficult to extend to simultaneously measure more than a dozen or so proteins; this limits their method's effectiveness in discovering realistic networks, which are much larger.

The method we propose deviates significantly from previous approaches. One of our contributions is the reformulation of the problem by invoking the Occam's Razor principle: our goal is the simplest explanation of the experimental data that is also consistent with the known biology of the signaling system. Surprisingly, this parsimony-based approach produces results that are quite plausible. We believe that this observation — that the Occam's Razor principle is useful in understanding signaling networks — is itself of significant importance. Another of our contributions is the explicit use of the current knowledge of the signaling network's structure to guide our search; in contrast, much of previous work has followed an *ab initio* approach. We observe that using the currently available information to guide the search significantly improves the quality of the results. Also, our method works for signaling networks of arbitrary sizes. It is also the first approach to combine RNAi data with PPI data for discovering signaling networks. From a computational perspective, one of our contributions is an information-flow based interpretation of signaling networks that allows us to make use of network flow algorithms from the theoretical graph analysis literature.

6.1 Problem Formulation

The input to the problem consists of:

1. The currently known topology of the chosen signaling subsystem, i.e., a directed graph N_0 consisting of nodes corresponding to known components of the signaling subsystem and directed edges corresponding to the known interactions. The most downstream node in N_0 should correspond to the end-effector gene T of the subsystem. We refer to N_0 as the *core cascade.*

2. A PPI network G. If confidence scores are available for the interactions in the PPI data, the edges may be weighted, i.e., $0 < w(e) \leq 1$ where e is an edge in G.

   To address the issues of poor coverage and quality in currently available PPI data, we combine PPI data from multiple species to construct our network. This combination is done using the mappings determined by IsoRank (§5), our algorithm for PPI network alignment. Furthermore, we use computational methods to predict PPIs from functional genomic data like gene co-expression, and by using a structure-based approach (Struct2Net, §3).

3. A set of RNAi hits $R = \{r_i\}$ and the corresponding scores $S = \{s_i\}$. The reporter gene of the RNAi experiment should be the same as the end-effector gene T of the chosen signaling subsystem.

Given these inputs, our goal is to produce a directed graph N^*, such that each node corresponds to a predicted (or known) component of the signaling network and each directed edge corresponds to an interaction. We require N^* to be consistent with the input data; in particular, we require the following constraints to be satisfied:

A1 All the nodes in N^* are present as RNAi hits (i.e. in R)

A2 Each edge in N^* is directed. Also, each directed arc $a \rightarrow b$ in N^* is either in N_0 or corresponds to an (undirected) edge a — b in G

A3 Every node in N^* has a directed path to the target gene T; T is thus the most downstream node in N^*

A4 Nodes closer to T should have higher RNAi scores. In particular, if N^* has an arc $a \rightarrow b$ where a and b are not part of N_0, then $s_a \leq s_b$. To allow for noise in RNAi scores, this inequality may be relaxed somewhat.

In the set of possible graphs that satisfy the above constraints, our desired output N^* is optimal under a weighted combination of the following objectives:

1. Maximize the number of nodes in N^*, i.e., try to explain as much of the data as possible.

2. Minimize the number of edges in N^*. Essentially, this is the parsimony requirement— the intuition here is that a sparse graph is a simpler explanation than a denser graph.

3. Maximize the agreement with known biological facts about the signaling subsystem. One of the key strengths of our proposed approach is its ability to formalize such biological knowledge as constraints on the structure of N^*. For example, the method can require that the structure of N^* give higher importance to RNAi hits that re-occur across various cell lines; encourage that for most genes X, its influence on the end-effector T be transmitted via a core-cascade of known components; and require that the topology of N^* be consistent with known epistasis data etc.

6.2 Brief Description of the Algorithm

Our algorithm works in two stages. In the first stage, we construct a directed graph which is consistent with the core cascade N_0, the input RNAi and PPI data and has T (the end-effector gene) as its most downstream node. In the second stage, we prune redundant edges from this graph, the goal being to find the sparsest graph — ideally, a directed tree — that explains the data and is also consistent with the available biological knowledge about the signaling system. We do this by constructing an integer linear program, relaxing it to a linear program and solving it.

Stage 1: We start by extracting the subgraph of PPI network composed only of RNAi hits. If a RNAi hit is not present in the PPI network, our method will not include it in the final output N^*. We also add in the nodes and edges from the core cascade N_0. We then impose directionality on the edges of this graph G_1. If a node X is a RNAi hit, we say that it influences the end-effector T. We argue that the output graph N^* should be such that influence flows along its edges, i.e., the edge direction should be in accordance with the pattern of influence flow.[2] In particular, we impose directionality on G_1 as follows. For each edge a — b in G_1:

[2]Because our base network is a PPI network, our assumption implies that influence will be transmitted by protein interactions. However, other kinds of influence mechanisms can be included by adding in an appropriate set of edges, if the relevant data is available.

- if a and b are both not in the core cascade N_0, then if $|s_a - s_b| > \epsilon$ then the edge direction is from b to a. If $|s_a - s_b| \leq \epsilon$ then the edge is bi-directed (i.e., its direction can not be reliably inferred in the first stage). Ideally, ϵ should be 0; however, to allow for noise in the RNAi scores, we set ϵ to a small positive value.

- if a is in the core cascade N_0 but not b, the edge direction is from b to a.

- if a and b are both in the core cascade N_0, the edge direction is the same as that specified in the core cascade N_0.

The above heuristics encode the following assumptions: (1) between nodes not in the core cascade, the influence flows from nodes with lower influence (i.e. RNAi score) to those with higher influence; (2) influence flow within the core cascade N_0 exactly matches the currently accepted understanding of the core cascade's structure; and (3) influence does not flow *out* of the core cascade. We believe that these assumptions represent a good trade-off, i.e, they are strong enough to constrain the possible search space yet flexible enough to allow most of the biologically plausible scenarios.

Given this directed graph G_2, we remove all its nodes (and incident edges) from which a directed path to T does not exist. Since our algorithm can not find for such nodes a path of influence flow to T, it does not include them in the final graph N^*.

Stage 2: After Stage 1, we have a graph G_2 that explains as much of the RNAi and PPI data as possible, under certain assumptions. We now search for the most parsimonious explanation by pruning redundant edges from G_2. Our goal is a directed spanning tree $G^\dagger$ of G_2. Intuitively, such a tree is the sparsest graph that still explains all the data that G_2 explains. However, there are many possible spanning trees of G_2 and we want the one which (1) gives primary importance to the core cascade, and (2) is the most consistent with other available biological information. Using ideas from multicommodity flow literature, we formulate an integer linear program (ILP) whose feasible space is the set of all possible directed spanning trees of G_2. We then tailor the objective function so that the optimal solution will correspond to the optimal tree under the above-mentioned goals. We relax this ILP to a linear program (LP) and solve it. Because of the relaxation and the presence of some bi-directed edges in G_2, our output graph N^* is not always a tree; however, it is almost always very similar to a tree.

We briefly describe some parts of the ILP we construct to find N^*. We start by creating an ILP whose feasible space consists of the spanning trees of G_2. This is done by constructing the following variant of a

classical multi-commodity flow problem (based on Magnanti and Wolsey [64]):

MC1 For each node X ($\neq T$), require that there be one unit of flow of type x from X to T.

MC2 Each edge has a capacity of one unit for each type of flow. Different types of flow can go together along the edge, as long as each is below one unit (this differs from a classical multi-commodity flow setup). For a uni-directional edge, the flow can only be along the edge's direction. For a bi-directed edge, the flow can be along either direction.

MC3 Denote an edge as "on" if there is a non-zero flow of any type along either direction of the edge. Require that exactly $n-1$ edges be "on", where n is the number of nodes in G_2.

Condition MC1 ensures that the feasible space consists of graphs where each node X has a directed path to T. Conditions MC2 and MC3 restrict this to the set of graphs with exactly $n-1$ edges. Together, they imply that the feasible space will consist of trees (a connected graph with n nodes and $n-1$ edges is always a tree). Formally, we write these constraints as part of an ILP:

$$\sum_{e\in\delta^-(T)} f_e^k - \sum_{e\in\delta^+(T)} f_e^k = 1 \quad \text{for all } k \in V_{G_2}, k \neq T \tag{6.1}$$

$$\sum_{e\in\delta^-(v)} f_e^k - \sum_{e\in\delta^+(v)} f_e^k = 0 \quad \text{for all } v \in V_{G_2}, v \neq T, v \neq k, \text{ and all } k \tag{6.2}$$

$$\sum_{e\in\delta^-(k)} f_e^k - \sum_{e\in\delta^+(k)} f_e^k = -1 \quad \text{for all } k \neq T \tag{6.3}$$

$$f_{ij}^k \leq y_{ij} \quad \text{for every arc } (i,j) \text{ and all } k \neq T \tag{6.4}$$

$$\sum_{e\in E_{G_2}} y_e = n-1 \tag{6.5}$$

$$f \geq 0 \tag{6.6}$$

$$y_e \in \{0,1\} \quad \text{for all arcs } e \in E_{G_2} \tag{6.7}$$

Here, V_{G_2} and E_{G_2} are the set of nodes and set of edges of G_2, respectively; $n = |V_{G_2}|$; $\delta^-(x)$ is the set of edges coming into node x; and $\delta^+(x)$ is the set of edges going out of x.

One biological conjecture that has received significant support is that the components of the core cascade are the main signal integrators in the signaling subsystem, i.e., the core cascade is the central trunk where signal from the peripheral nodes is integrated [36, 35]. This conjecture is supported by the observation that, across various conditions, the core cascade components consistently turn out to be

among the strongest influencers of the end-effector T. We encode this intuition by requiring the following structure in the output: given a node x with two paths to T such that one passes through the core cascade and the other does not, we prefer a tree that contains the former. This requirement can be elegantly incorporated into our flow-based formulation:

$$\text{maximize } d_z$$

subject to these additional constraints:

$$z_x = \sum_{k \in V_{G_2}, k \neq T} \sum_{e \in \delta^-(x)} f_e^k \tag{6.8}$$

$$z^* = \sum_r z_r \text{ for all } r \in V_{N_0} \text{ such that } (r, T) \in E_{G_2} \tag{6.9}$$

$$z^* - z_x \geq d_z \text{ for all } x \in V_{G_2}, x \ni V_{N_0} \tag{6.10}$$

$$\tag{6.11}$$

where V_{N_0} is the set of nodes in the core cascade. Here, we first compute the total flow z_x coming into a node x. By our previous construction, this is exactly the number of paths that go through x. We then compute z^*, the total number of paths to T via any of the core cascade nodes, and maximize the difference between it and the flow z_x through any non-core node.

Similarly, we can add more constraints and terms to the objective function to encode more biological information. For example, if epistatis data is available and indicates that a node x is upstream of a node y, we could encode a preference for solutions where the path from x to T goes via y. Many other biological constraints can also be expressed.

6.3 Results: Exploring the MAPK Cascade

We describe here a simple test case for the algorithm. The constraints imposed on the structure of N^* are quite simple and biologically intuitive; yet, the inferred influence flow network contains surprisingly plausible hypotheses.

As a first test, we supplied only a part of the known MAPK cascade (in fly) to the method and tested what its predictions were about the remaining core cascade nodes. The core MAPK cascade is *Drk* $\rightarrow$

Sos $\rightarrow$ *Ras85D* $\rightarrow$ *Raf* $\rightarrow$ *Dsor1* $\rightarrow$ *Erk*. For test purposes, we specified to our method only a truncated cascade consisting of *Raf, Dsor1* and *Erk* (see Fig 6-1). Our method was able to retrieve all the remaining core nodes (*Drk, Sos, Ras85D*). Furthermore, *Ras85D* and *Sos* were two of the three nodes with the most flow in our Linear Program's solution (in our method, this quantity is a proxy for the node's importance in the solution). Fig 6-2 shows the output when the entire cassette is specified.

Next, we supplied the entire core cascade as part of the problem input and looked at nodes that were shown to have high flow going through them. One of these is $14\_3\_3\zeta$. This protein has been documented to help differentially regulate the MAPK pathway [41]. Another node highlighted by our analysis is Myb. It has been postulated to be involved in pathways that regulate cell size and cell cycle progression [13].

6.4 Future Work

We have described a parsimony-based method for producing hypotheses about a specific signaling network's topology by combining PPI and RNAi data and making judicious use of available biological data. We construct a linear program whose solution corresponds to a sparse directed graph that is consistent with the available data as well as the current understanding of the signaling network's structure.

One of the key contributions of our method is a flow-based computational formulation that mirrors the biological intuition of information flow in a signaling network. It captures, in a very natural way, much of biological knowledge and conjectures regarding the mechanism of influence transmission within such networks. Also, we have obtained some success in applying a parsimony-based approach to network discovery. This suggests that other parsimony-based approaches may also be attempted.

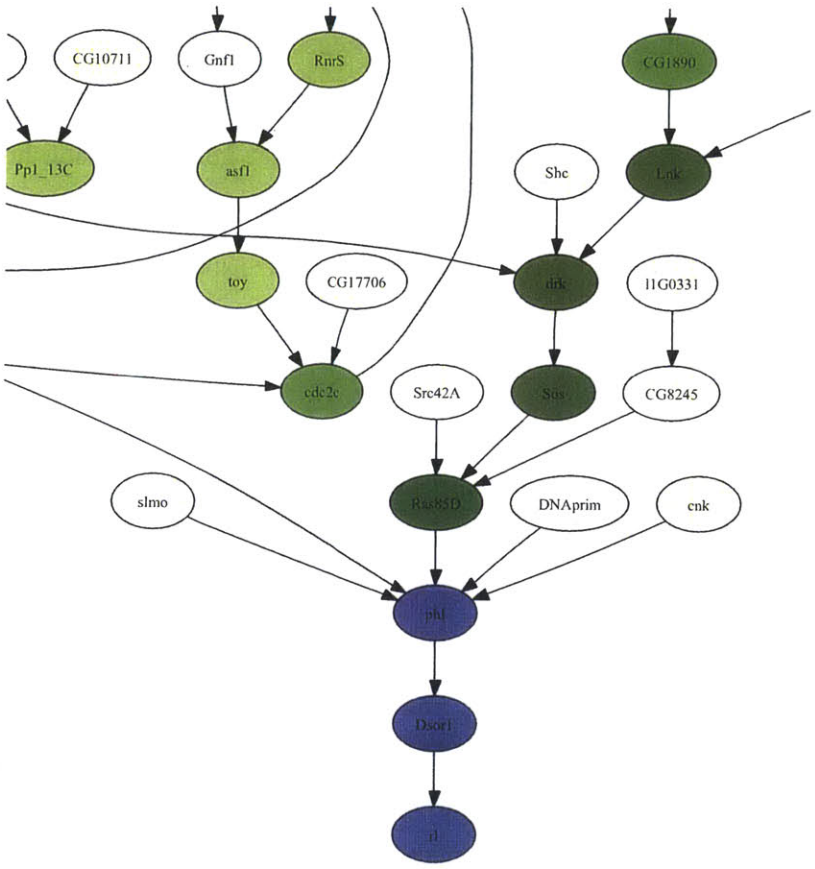


Figure 6-1: **Part of the output graph when a truncated cascade is supplied:** We show here a part of the output graph N^* when a truncated MAPK cascade is specified to the algorithm. The actual MAPK core cascade in fly is *Drk* $\rightarrow$ *Sos* $\rightarrow$ *Ras* $\rightarrow$ *Phl* $\rightarrow$ *Dsor1* $\rightarrow$ *Erk* (*Rl* and *Erk* refer to the same gene). When specified only a part of this cascade (blue nodes), the algorithm was able to retrieve the remaining nodes, along with the correct set of connections. Furthermore, the dark green color of these nodes indicates that they have high z_x values (Eqn 6.8), i.e., a lot of paths to *Erk* go through them. This suggests that our algorithm assigned higher importance to them.

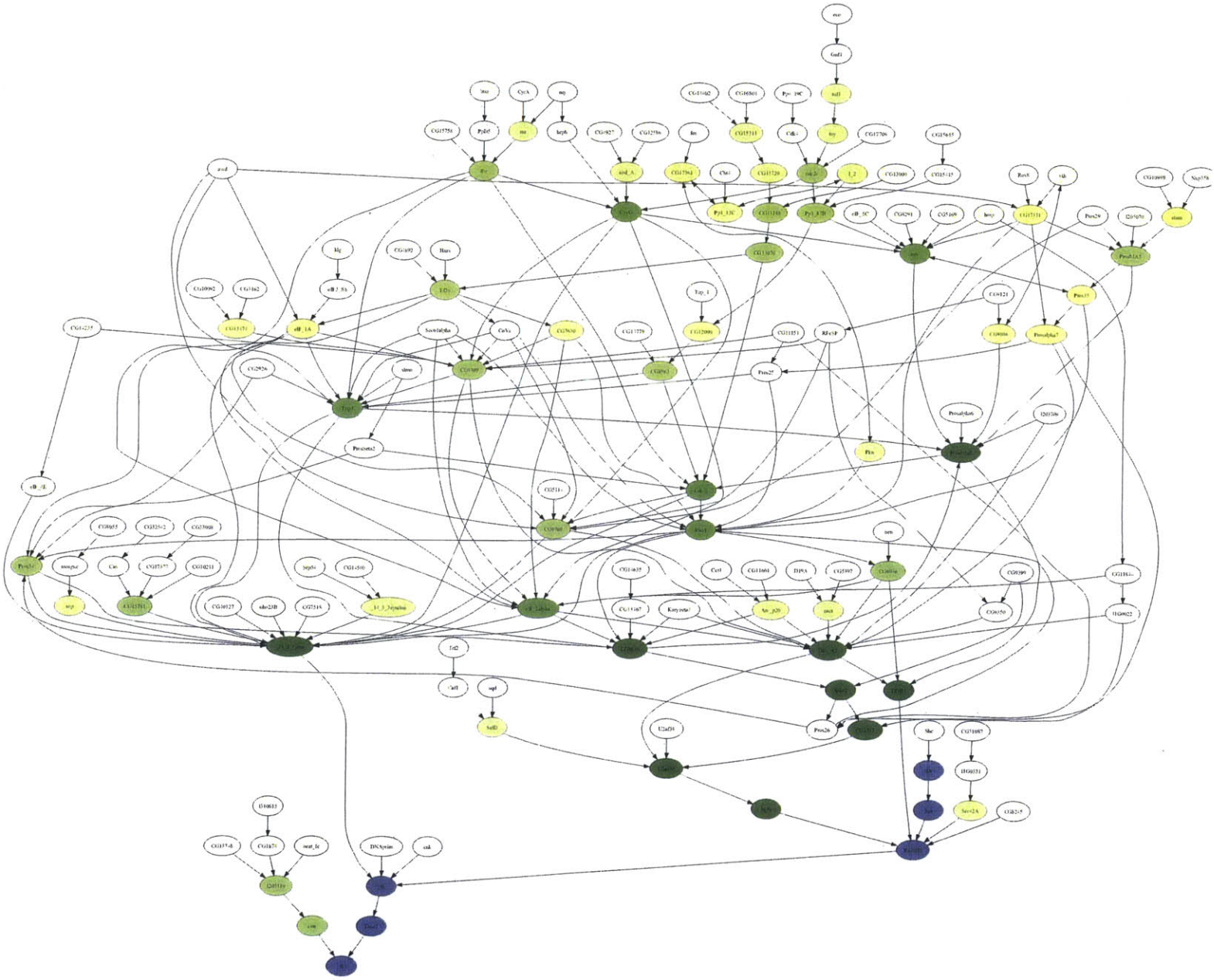


Figure 6-2: **Output graph generated for positive regulators of the MAPK signaling network:** We show here the output graph N^* corresponding to positive regulators (as identified by RNAi experiments [35]) of the MAPK signaling subsystem. The blue nodes are the components of the known MAPK cascade; the bottom-most node is *Rl* (i.e., *Erk*), the end-effector of the subsystem. For other nodes, darker colors indicate higher z_x values (Eqn 6.8) and imply that a lot of paths to *Erk* are routed through that node; we interpreted this as a proxy of the node's importance in the network.

Chapter 7

Conclusion

In this thesis, we have discussed various algorithms for the analysis of PPI data. While these algorithms have been presented separately in the preceding chapters, they were designed to inter-operate with each other as part of an overall system. In this chapter, we discuss how they fit together and what future work is needed to produce a coherent system of PPI data acquisition and analysis.

7.1 Towards a System of PPI Data Acquisition and Analysis

Like most biological datasets, the "lifecycle" of PPI data consists of three parts: data collection, data cleanup and data analysis (Fig 7-1). Computational techniques can be a part of any of these stages.

Computational techniques for PPI data acquisition can be used to enable an experimental protocol or increase its efficacy; for example, Bandeira *et al.* have described computational approaches for *de novo* peptide sequencing by mass-spectrometry when performing co-immunoprecipitation assays [5]. Alternatively, computational approaches can be used to predict novel PPIs. In Chap. 3 we describe such an approach, where we predict PPIs using structure-based insights, along with other functional genomic data. The predictions of purely computational methods can be used either to direct biological assays or combined with experimental PPI data to increase coverage. The combination process may involve an error model to express our relative confidence in the various PPI sources.

Given PPIs from various sources, a computational error model can be very useful in combining the datasets and distinguishing between PPIs with varying levels of biological plausibility. Sometimes the data acquisition and error-modeling steps may be combined. For example, with some biological assays

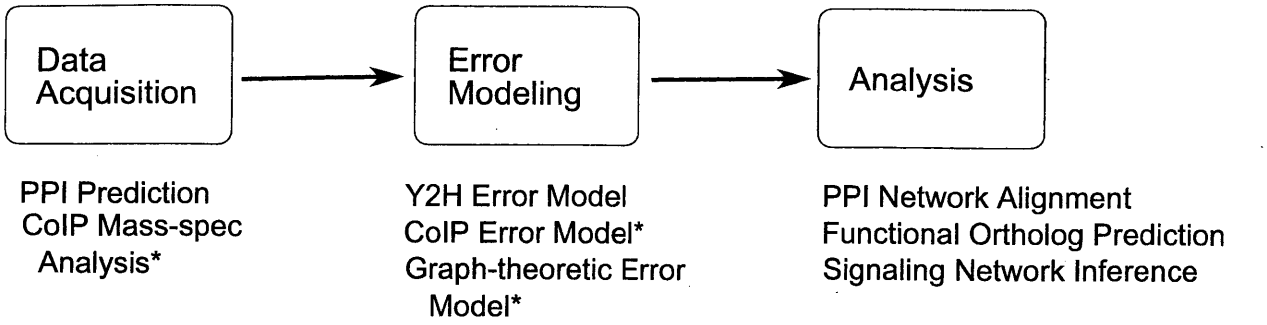


Figure 7-1: **A System for Analyzing PPI Data:** We describe the three main stages in PPI data analysis where computational techniques may be involved. Below each stage are listed key computational analyses relevant to that stage. Most of these analyses have been described in the preceding chapters of this thesis; the ones marked with an asterisk are candidates for future work.

for discovering PPIs, one can get information about how often a given PPI appeared in repeated trials, providing a direct quantitative measure [48]. Our work in Chap. 4 describes another approach to identifying errors in experimental data, by creating a Bayesian error model for data from Yeast 2-Hybrid assays. A similar model can be designed for co-immunoprecipitation data. Finally, one can combine these piecewise error models into a more comprehensive model. For example, the STRING database [84] combines experimental and computational PPI data into a single dataset, using machine learning approaches. A key challenge in this sub-domain is the need for a good "gold-standard" set of positive and negative examples of PPIs, which can be used to train such models.

Once a well-cleaned PPI dataset is available, one can use computational analysis of PPI data to gain biological insights. Such analyses may span a wide variety. There are analyses that look only the PPI data and, by graph-theoretic analysis of the PPI network, gain insights into the cellular system [52]. More commonly, PPI data is combined with other biological datasets for integrative analysis. In this thesis, we have described two such analyses. In Chap. 5 we described an algorithm that combines PPI and sequence data for comparative genomics of PPI networks. As a result of the analysis, we are able to infer functional orthologs, which better capture gene correspondences across species. In Chap. 6, we describe an analysis where we combine PPI data with RNA-interference data to better understand signaling networks. Various other analyses have also been described: predicting function using PPI networks and GO terms [68], integrating PPI and expression data [42] and so on.

In Fig 7-1, we depict the data-lifecycle of PPI data and some examples of computational methods that might be involved in each stage. Many of the listed examples correspond to algorithms described in this

thesis. We have marked the remaining methods with asterisks; these would be good candidates for future work. Together, these methods can be part of a PPI-analysis system that can be used by a researcher for data acquisition, error-modeling, and, finally, analytics.

Bibliography

[1] *http://chianti.ucsd.edu/NetworkBlast.*

[2] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using pagerank vectors. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 475–486, 2006.

[3] Antonina Andreeva, Dave Howorth, Steven E. Brenner, Tim J. Hubbard, Cyrus Chothia, and Alexey G. Murzin. SCOP database in 2004: refinements integrate structure and sequence family data. *Nucleic acids research*, 32(Database issue):D226–D229, January 2004.

[4] J. S. Bader, A. Chaudhuri, J. M. Rothberg, and J. Chant. Gaining confidence in high-throughput protein interaction networks. *Nature biotechnology*, 22(1):78–85, 2004.

[5] N. Bandeira, D. Tsur, A. Frank, and P. Pevzner. Protein identification by spectral network analysis. *Proceedings of the National Academy of Sciences*, 104(15):6140–6145, 2007.

[6] S. Bandyopadhyay, R. Sharan, and T. Ideker. Systematic identification of functional orthologs based on protein network comparison. *Genome research*, 16(3):428–435, 2006.

[7] David P. Bartel and Chang-Zheng Z. Chen. Micromanagers of gene expression: the potentially widespread influence of metazoan microRNAs. *Nature reviews. Genetics*, 5(5):396–400, May 2004.

[8] G. I. Bell and K. S. Polonsky. Diabetes mellitus and genetically programmed defects in beta-cell function. *Nature*, 414(6865):788–791, December 2001.

[9] Johannes Berg and Michael Lässig. Cross-species analysis of biological networks by Bayesian alignment. *Proceedings of the National Academy of Sciences*, 103(29):10967–10972, July 2006.

[10] Bonnie Berger and Tom Leighton. Protein folding in the hydrophobic-hydrophilic(HP) model is NP-complete. In *RECOMB '98: Proceedings of the second annual international conference on Computational molecular biology*, pages 30–39, New York, NY, USA, 1998. ACM Press.

[11] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic acids research*, 28(1):235–242, January 2000.

[12] E. Birney, D. Andrews, M. Caccamo, Y. Chen, L. Clarke, G. Coates, T. Cox, F. Cunningham, V. Curwen, T. Cutts, T. Down, R. Durbin, X. M. Fernandez-Suarez, P. Flicek, S. Graf, M. Hammond, J. Herrero, K. Howe, V. Iyer, K. Jekosch, A. Kahari, A. Kasprzyk, D. Keefe, F. Kokocinski, E. Kulesha, D. London, I. Longden, C. Melsopp, P. Meidl, B. Overduin, A. Parker, G. Proctor, A. Prlic, M. Rae, D. Rios, S. Redmond, M. Schuster, I. Sealy, S. Searle, J. Severin, G. Slater, D. Smedley, J. Smith, A. Stabenau, J. Stalker, S. Trevanion, A. Ureta-Vidal, J. Vogel, S. White, C. Woodwark, and T. J. Hubbard. Ensembl 2006. *Nucleic acids research*, 34(Database issue):D556–61, 2006.

[13] Mikael Bjã¶rklund, Minna Taipale, Markku Varjosalo, Juha Saharinen, Juhani Lahdenperã, and Jussi Taipale. Identification of pathways regulating cell size and cell-cycle progression by RNAi. *Nature*, 439(7079):1009–1013.

[14] R. Bonneau, J. Tsai, I. Ruczinski, D. Chivian, C. Rohl, C. E. Strauss, and D. Baker. Rosetta in CASP4: progress in ab initio protein structure prediction. *Proteins*, Suppl 5:119–126, 2001.

[15] Philip Bradley, Kira M. Misura, and David Baker. Toward high-resolution de novo structure prediction for small proteins. *Science (New York, N.Y.)*, 309(5742):1868–1871, September 2005.

[16] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001.

[17] L. Chang and M. Karin. Mammalian MAP kinase signalling cascades. *Nature*, 410(6824):37–40, 2001.

[18] J. M. Claverie. Gene number. What if there are only 30,000 human genes? *Science (New York, N.Y.)*, 291(5507):1255–1257, February 2001.

[19] Guy R. Cochrane and Michael Y. Galperin. The 2010 Nucleic Acids Research Database Issue and online Database Collection: a community of data resources. *Nucleic Acids Research*, 38(suppl 1):D1–D4, January 2010.

[20] FlyBase Consortium. The FlyBase database of the Drosophila genome projects and community literature. *Nucleic acids research*, 31(1):172–175, 2003.

[21] Michael E. Cusick, Niels Klitgord, Marc Vidal, and David E. Hill. Interactome: gateway into systems biology. *Human Molecular Genetics*, 14(suppl 2):R171–R181, October 2005.

[22] R. DasGupta, A. Kaykas, R. T. Moon, and N. Perrimon. Functional genomic analysis of the Wnt-wingless signaling pathway. *Science*, 308(5723):826–833, 2005.

[23] Minghua Deng, Shipra Mehta, Fengzhu Sun, and Ting Chen. Inferring DomainDomain Interactions From ProteinProtein Interactions. *Genome Research*, 12(10):1540–1548, October 2002.

[24] T. E. Dever. Gene-specific regulation by general translation factors. *Cell*, 108(4):545–556, February 2002.

[25] D. B. Doroquez and I. Rebay. Signal integration during development: mechanisms of EGFR and Notch pathway function and cross-talk. *Critical reviews in biochemistry and molecular biology*, 41(6):339–385, 2006.

[26] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2 edition, November 2001.

[27] Gerard I. Evan and Karen H. Vousden. Proliferation, cell cycle and apoptosis in cancer. *Nature*, 411(6835):342–348, May 2001.

[28] Rob M. Ewing, Peter Chu, Fred Elisma, Hongyan Li, Paul Taylor, Shane Climie, Linda McBroom-Cerajewski, Mark D. Robinson, Liam O'Connor, Michael Li, Rod Taylor, Moyez Dharsee, Yuen Ho, Adrian Heilbut, Lynda Moore, Shudong Zhang, Olga Ornatsky, Yury V. Bukhman, Martin Ethier, Yinglun Sheng, Julian Vasilescu, Mohamed Abu-Farha, Jean-Philippe P. Lambert, Henry S. Duewel, Ian I. Stewart, Bonnie Kuehl, Kelly Hogue, Karen Colwill, Katharine Gladwish, Brenda Muskat, Robert Kinach, Sally-Lin L. Adams, Michael F. Moran, Gregg B. Morin, Thodoros Topaloglou, and Daniel Figeys. Large-scale mapping of human protein-protein interactions by mass spectrometry. *Molecular systems biology*, 3, 2007.

[29] J. Y. Fang and B. C. Richardson. The MAPK signalling pathways and colorectal cancer. *The lancet oncology*, 6(5):322–327, 2005.

[30] C. Fields, M. D. Adams, O. White, and J. C. Venter. How many genes in the human genome? *Nat Genet*, 7(3):345–346, July 1994.

[31] S. Fields and O. Song. A novel genetic system to detect protein-protein interactions. *Nature*, 340(6230):245–246, July 1989.

[32] A. Fire, S. Xu, M. K. Montgomery, S. A. Kostas, S. E. Driver, and C. C. Mello. Potent and specific genetic interference by double-stranded RNA in Caenorhabditis elegans. *Nature*, 391(6669):806–811, February 1998.

[33] J. Flannick, A. Novak, B. S. Srinivasan, H. H. McAdams, and S. Batzoglou. Graemlin: general and robust alignment of multiple large interaction networks. *Genome research*, 16(9):1169–1181, 2006.

[34] Jessica Fong, Amy Keating, and Mona Singh. Predicting specificity in bZIP coiled-coil protein interactions. *Genome Biology*, 5(2):R11+, 2004.

[35] A. Friedman and N. Perrimon. A functional RNAi screen for regulators of receptor tyrosine kinase and ERK signalling. *Nature*, 444(7116):230–234, 2006.

[36] A. Friedman and N. Perrimon. Genetic screening for signal transduction in the era of network biology. *Cell*, 128(2):225–231, 2007.

[37] Adam Friedman and Norbert Perrimon. A functional RNAi screen for regulators of receptor tyrosine kinase and ERK signalling. *Nature*.

[38] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman, 1979.

[39] Anne-Claude Gavin, Markus Bosche, Roland Krause, Paola Grandi, Martina Marzioch, Andreas Bauer, Jorg Schultz, Jens M. Rick, Anne-Marie Michon, Cristina-Maria Cruciat, Marita Remor, Christian Hofert, Malgorzata Schelder, Miro Brajenovic, Heinz Ruffner, Alejandro Merino, Karin Klein, Manuela Hudak, David Dickson, Tatjana Rudi, Volker Gnau, Angela Bauch, Sonja Bastuck, Bettina Huhse, Christina Leutwein, Marie-Anne Heurtier, Richard R. Copley, Angela Edelmann, Erich Querfurth, Vladimir Rybin, Gerard Drewes, Manfred Raida, Tewis Bouwmeester, Peer Bork,

Bertrand Seraphin, Bernhard Kuster, Gitte Neubauer, and Giulio Superti-Furga. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141–147, January 2002.

[40] Yanzhi Guo, Lezheng Yu, Zhining Wen, and Menglong Li. Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences. *Nucleic acids research*, 36(9):3025–3030, May 2008.

[41] Xing H., Zhang S., Weinheimer C., Kovacs A., and Muslin A.J. 14-3-3 proteins block apoptosis and differentially regulate mapk cascades. *EMBO J.*, 19(3):349–58, February 2000.

[42] J. D. Han, N. Bertin, T. Hao, D. S. Goldberg, G. F. Berriz, L. V. Zhang, D. Dupuy, A. J. Walhout, M. E. Cusick, F. P. Roth, and M. Vidal. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 430(6995):88–93, 2004.

[43] T. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman and Hall, 1990.

[44] J. Hirosumi, G. Tuncman, L. Chang, C. Z. Gorgun, K. T. Uysal, K. Maeda, M. Karin, and G. S. Hotamisligil. A central role for JNK in obesity and insulin resistance. *Nature*, 420(6913):333–336, 2002.

[45] Oliver Hobert. Gene Regulation by Transcription Factors and MicroRNAs. *Science*, 319(5871):1785–1786, March 2008.

[46] International Human Genome Sequencing Consortium. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945, October 2004.

[47] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences of the United States of America*, 98(8):4569–4574, April 2001.

[48] T. Ito, K. Tashiro, S. Muta, R. Ozawa, T. Chiba, M. Nishizawa, K. Yamamoto, S. Kuhara, and Y. Sakaki. Toward a protein-protein interaction map of the budding yeast: A comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proceedings of the National Academy of Sciences*, 97(3):1143–1147, 2000.

[49] Ariel Jaimovich, Gal Elidan, Hanah Margalit, and Nir Friedman. Towards an Integrated Protein-Protein Interaction Network: A Relational Markov Network Approach. *Journal of Computational Biology*, 13(2):145–164, March 2006.

[50] Kevin A. Janes and Michael B. Yaffe. Data-driven modelling of signal-transduction networks. *Nature Reviews Molecular Cell Biology*, 7(11):820–828, November 2006.

[51] Ronald Jansen, Haiyuan Yu, Dov Greenbaum, Yuval Kluger, Nevan J. Krogan, Sambath Chung, Andrew Emili, Michael Snyder, Jack F. Greenblatt, and Mark Gerstein. A Bayesian Networks Approach for Predicting Protein-Protein Interactions from Genomic Data. *Science*, 302(5644):449–453, October 2003.

[52] H. Jeong, S. P. Mason, A. L. Barabasi, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–42, 2001.

[53] J. D. Jordan, E. M. Landau, and R. Iyengar. Signaling networks: the origins of cellular multitasking. *Cell*, 103(2):193–200, 2000.

[54] B. P. Kelley, R. Sharan, R. M. Karp, T. Sittler, D. E. Root, B. R. Stockwell, and T. Ideker. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proceedings of the National Academy of Sciences*, 100(20):11394–11399, 2003.

[55] M. Kellis, N. Patterson, B. Birren, B. Berger, and E. S. Lander. Methods in comparative genomics: genome correspondence, gene identification and regulatory motif discovery. *Journal of computational biology*, 11(2-3):319–355, 2004.

[56] H. Kitano. Computational systems biology. *Nature*, 420(6912):206–210, 2002.

[57] Hiroaki Kitano. Systems Biology: A Brief Overview. *Science*, 295(5560):1662–1664, March 2002.

[58] M. Koyuturk, Y. Kim, S. Subramaniam, W. Szpankowski, and A. Grama. Detecting conserved interaction patterns in biological networks. *Journal of computational biology*, 13(7):1299–1322, 2006.

[59] Nevan J. Krogan, Gerard Cagney, Haiyuan Yu, Gouqing Zhong, Xinghua Guo, Alexandr Ignatchenko, Joyce Li, Shuye Pu, Nira Datta, Aaron P. Tikuisis, Thanuja Punna, JosÃ© M. PeregrÃn

Alvarez, Michael Shales, Xin Zhang, Michael Davey, Mark D. Robinson, Alberto Paccanaro, James E. Bray, Anthony Sheung, Bryan Beattie, Dawn P. Richards, Veronica Canadien, Atanas Lalev, Frank Mena, Peter Wong, Andrei Starostine, Myra M. Canete, James Vlasblom, Samuel Wu, Chris Orsi, Sean R. Collins, Shamanta Chandran, Robin Haw, Jennifer J. Rilstone, Kiran Gandi, Natalie J. Thompson, Gabe Musso, Peter St Onge, Shaun Ghanny, Mandy H. Y. Lam, Gareth Butland, Amin M. Altaf-Ul, Shigehiko Kanaya, Ali Shilatifard, Erin O'Shea, Jonathan S. Weissman, C. James Ingles, Timothy R. Hughes, John Parkinson, Mark Gerstein, Shoshana J. Wodak, Andrew Emili, and Jack F. Greenblatt. Global landscape of protein complexes in the yeast Saccharomyces cerevisiae. *Nature*, 440(7084):637–643, March 2006.

[60] Y. F. Leung and D. Cavalieri. Fundamentals of cDNA microarray data analysis. *Trends Genet*, 19(11):649–659, November 2003.

[61] N. Lin, B. Wu, R. Jansen, M. Gerstein, and H. Zhao. Information assessment on predicting protein-protein interactions. *BMC bioinformatics*, 5:154, 2004.

[62] David J. Lunn, Andrew Thomas, Nicky Best, and David Spiegelhalter. WinBUGS - A Bayesian modelling framework: Concepts, structure, and extensibility. *Statistics and Computing*, 10(4):325–337, October 2000.

[63] Ian J. MacRae, Kaihong Zhou, Fei Li, Adrian Repic, Angela N. Brooks, W. Zacheus Cande, Paul D. Adams, and Jennifer A. Doudna. Structural Basis for Double-Stranded RNA Processing by Dicer. *Science*, 311(5758):195–198, January 2006.

[64] T. L. Magnanti and L. A. Wolsey. Optimal Trees. *MIT Operations Research Center Working Papers*, OR-290-94, 1994.

[65] Sergei Maslov and Kim Sneppen. Specificity and Stability in Topology of Protein Networks. *Science*, 296(5569):910–913, May 2002.

[66] H. W. Mewes, D. Frishman, U. Güldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Münsterkötter, S. Rudd, and B. Weil. MIPS: a database for genomes and protein sequences. *Nucleic acids research*, 30(1):31–34, January 2002.

[67] G. R. Mishra, M. Suresh, K. Kumaran, N. Kannabiran, S. Suresh, P. Bala, K. Shivakumar, N. Anuradha, R. Reddy, T. M. Raghavan, S. Menon, G. Hanumanthu, M. Gupta, S. Upendran, S. Gupta,

M. Mahesh, B. Jacob, P. Mathew, P. Chatterjee, K. S. Arun, S. Sharma, K. N. Chandrika, N. Deshpande, K. Palvankar, R. Raghavnath, R. Krishnakanth, H. Karathia, B. Rekha, R. Nayak, G. Vishnupriya, H. G. Kumar, M. Nagini, G. S. Kumar, R. Jose, P. Deepthi, S. S. Mohan, T. K. Gandhi, H. C. Harsha, K. S. Deshpande, M. Sarker, T. S. Prasad, and A. Pandey. Human protein reference database–2006 update. *Nucleic acids research*, 34(Database issue), January 2006.

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

[69] Oaz Nir, Chris Bakal, Norbert Perrimon, and Bonnie Berger. Inference of RhoGAP/GTPase regulation using single-cell morphological data from a combinatorial RNAi screen. *Genome Research*, 20(3):372–380, March 2010.

[70] C. Papadimitriou and K. Steiglitz. *Combinatorial optimization*. Dover, 1998.

[71] J. A. Papin, T. Hunter, B. O. Palsson, and S. Subramaniam. Reconstruction of cellular signalling networks and analysis of their properties. *Nature reviews.Molecular cell biology*, 6(2):99–111, 2005.

[72] J. A. Papin and B. O. Palsson. The JAK-STAT signaling network in the human B-cell: an extreme signaling pathway analysis. *Biophysical journal*, 87(1):37–46, 2004.

[73] J. A. Papin, N. D. Price, S. J. Wiback, D. A. Fell, and B. O. Palsson. Metabolic pathways in the post-genome era. *Trends in biochemical sciences*, 28(5):250–258, 2003.

[74] P. Polakis. The many ways of Wnt in cancer. *Current opinion in genetics & development*, 17(1):45–51, 2007.

[75] Kannanganattu V. Prasanth and David L. Spector. Eukaryotic regulatory RNAs: an answer to the 'genome complexity' conundrum. *Genes & Development*, 21(1):11–42, January 2007.

[76] Y. Qi, Z. Bar-Joseph, and J. Klein-Seetharaman. Evaluation of different biological data and computational classification methods for use in protein interaction prediction. *Proteins*, 63(3):490–500, 2006.

[77] M. Remm, C. E. Storm, and E. L. Sonnhammer. Automatic clustering of orthologs and in-paralogs from pairwise species comparisons. *Journal of Molecular Biology*, 314(5):1041–1052, 2001.

[78] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

[79] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science (New York, N.Y.)*, 270(5235):467–470, October 1995.

[80] Judith S. Sebolt-Leopold and Roman Herrera. Targeting the mitogen-activated protein kinase cascade to treat cancer. *Nature Reviews Cancer*, 4(12):937–947, December 2004.

[81] R. Sharan, S. Suthram, R. M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R. M. Karp, and T. Ideker. Conserved patterns of protein interaction in multiple species. *Proceedings of the National Academy of Sciences*, 102(6):1974–1979, 2005.

[82] K. T. Simons, C. Kooperberg, E. Huang, and D. Baker. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *Journal of molecular biology*, 268(1):209–225, April 1997.

[83] C. Stark, B. J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. BioGRID: a general repository for interaction datasets. *Nucleic acids research*, 34(Database issue):D535–9, 2006.

[84] Damian Szklarczyk, Andrea Franceschini, Michael Kuhn, Milan Simonovic, Alexander Roth, Pablo Minguez, Tobias Doerks, Manuel Stark, Jean Muller, Peer Bork, Lars J. Jensen, and Christian Mering. The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Research*, 39(suppl 1):D561–D568, January 2011.

[85] J. Taipale and P. A. Beachy. The Hedgehog and Wnt signalling pathways in cancer. *Nature*, 411(6835):349–354, May 2001.

[86] R. L. Tatusov, M. Y. Galperin, D. A. Natale, and E. V. Koonin. The COG database: a tool for genome-scale analysis of protein functions and evolution. *Nucleic acids research*, 28(1):33–36, January 2000.

[87] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, and J. M. Rothberg. A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. *Nature*, 403(6770):623–627, 2000.

[88] Peter Uetz, Yu A. Dong, Christine Zeretzke, Christine Atzler, Armin Baiker, Bonnie Berger, Seesandra V. Rajagopala, Maria Roupelieva, Dietlind Rose, Even Fossum, and Jurgen Haas. Herpesviral Protein Networks and Their Interaction with the Human Proteome. *Science*, 311(5758):239–242, January 2006.

[89] P. O. Vidalain, M. Boxem, H. Ge, S. Li, and M. Vidal. Increasing specificity in high-throughput yeast two-hybrid experiments. *Methods*, 32(4):363–370, 2004.

[90] C. von Mering, R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields, and P. Bork. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417(6887):399–403, 2002.

[91] Haidong Wang, Eran Segal, Asa Ben-Hur, Qianru Li, Marc Vidal, and Daphne Koller. InSite: a computational method for identifying protein-protein interaction binding sites on a proteome-wide scale. *Genome Biology*, 8:R192+, September 2007.

[92] I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S. M. Kim, and D. Eisenberg. DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucleic acids research*, 30(1):303–305, 2002.

[93] J. Xu, M. Li, D. Kim, and Y. Xu. RAPTOR: optimal protein threading by linear programming. *J Bioinform Comput Biol*, 1(1):95–117, April 2003.

[94] Jinbo Xu and Ming Li. Assessment of RAPTOR's linear programming approach in CAFASP3. *Proteins*, 53(S6):579–584, 2003.

[95] C. H. Yeang, T. Ideker, and T. Jaakkola. Physical network models. *Journal of computational biology*, 11(2-3):243–262, 2004.

[96] C. H. Yeang, H. C. Mak, S. McCuine, C. Workman, T. Jaakkola, and T. Ideker. Validation and refinement of gene-regulatory pathways on a network of physical interactions. *Genome biology*, 6(7):R62, 2005.

[97] P. Zamore, T. Tuschl, P. Sharp, and D. Bartel. RNAiDouble-Stranded RNA Directs the ATP-Dependent Cleavage of mRNA at 21 to 23 Nucleotide Intervals. *Cell*, 101(1):25–33, March 2000.

[98] P. D. Zamore. RNA interference: big applause for silencing in Stockholm. *Cell*, 127(6):1083–1086, December 2006.

[99] L. V. Zhang, S. L. Wong, O. D. King, and F. P. Roth. Predicting co-complexed protein pairs using genomic and proteomic data integration. *BMC bioinformatics*, 5:38, 2004.