Bridging Methodological Gaps in Network-Based Systems Biology

Christopher L. Poirel

Dissertation submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science

T. M. Murali, Chair
Ananth Grama
Narendran Ramakrishnan
John Tyson
Anil Vullikanti

August 30, 2013
Blacksburg, Virginia

Keywords: Computational Biology, Functional Enrichment, Graph Theory, Network, Random Walk, Signaling Pathways, Top-Down Analysis

Bridging Methodological Gaps in Network-Based Systems Biology

Christopher L. Poirel

(ABSTRACT)

Functioning of the living cell is controlled by a complex network of interactions among genes, proteins, and other molecules. A major goal of systems biology is to understand and explain the mechanisms by which these interactions govern the cell's response to various conditions. Molecular interaction networks have proven to be a powerful representation for studying cellular behavior. Numerous algorithms have been developed to unravel the complexity of these networks. Our work addresses the drawbacks of existing techniques. *This thesis includes three related research efforts that introduce network-based approaches to bridge current methodological gaps in systems biology.*

i. Functional enrichment methods provide a summary of biological functions that are overrepresented in an interesting collection of genes (e.g., highly differentially expressed genes between a diseased cell and a healthy cell). Standard functional enrichment algorithms ignore the known interactions among proteins. We propose a novel network-based approach to functional enrichment that explicitly accounts for these underlying molecular interactions. Through this work, we close the gap between set-based functional enrichment and topological analysis of molecular interaction networks.

ii. Many techniques have been developed to compute the response network of a cell. A recent trend in this area is to compute response networks of small size, with the rationale that only part of a pathway is often changed by disease and that interpreting small subnetworks is easier than interpreting larger ones. However, these methods may not uncover the spectrum of pathways perturbed in a particular experiment or disease. To avoid these difficulties, we propose to use algorithms that reconcile case-control DNA microarray data with a molecular interaction network by modifying per-gene differential expression p-values such that two genes connected by an interaction show similar changes in their gene expression values.

iii. Top-down analyses in systems biology can automatically find correlations among genes and proteins in large-scale datasets. However, it is often difficult to design experiments from these results. In contrast, bottom-up approaches painstakingly craft detailed models of cellular processes. However, developing the models is a manual process that can take many years. These approaches have largely been developed independently. We present LINKER, an efficient and automated data-driven method that analyzes molecular interactomes. LINKER combines teleporting random walks and k-shortest path computations to discover connections from a set of source proteins to a set of target proteins. We demonstrate the efficacy of LINKER through two applications: proposing extensions to an existing model of cell cycle regulation in budding yeast and automated reconstruction of human signaling pathways. LINKER achieves superior precision and recall compared to state-of-the-art algorithms from the literature.

Dedicated to the memory of my late grandfather
Weston "Posh" Stevens
for providing a lifetime of love, leadership, and laughter.

Acknowledgments

My deepest appreciation goes to T. M. Murali for inviting me to join his research group at Virginia Tech (VT) in 2009. Murali exposed me to the wonderful world of systems biology and the exciting open graph-theoretical challenges therein. His philosophy of teaching by example has greatly impacted my research. Through his mentorship, Murali demonstrated that careful attention to detail is necessary to accomplish high quality, reproducible research. I am greatly appreciative of his persistent words of encouragement and sound advice.

Working with Murali introduced me to John Tyson. John's encyclopedic knowledge of dynamic systems (especially in yeast) has been an indispensable resource. His biological modeling expertise provided a fresh perspective to the challenging problems tackled in this thesis, and conversations with John had a profound influence on the algorithms we developed.

Naren's discussions of machine learning and knowledge discovery have greatly impacted my development and enthusiasm as a researcher. His presentation of difficult concepts in data mining have been incredibly inspiring. Naren's data mining seminars and open discussion teaching style were wonderful experiences and helped me to explore some of my own research questions with new approaches.

When I first joined the PhD program at VT, I had the pleasure of working with Anil as a graduate teaching assistant for his course on theory of algorithms. Anil's expertise in graph theory and algorithm analysis was a vital resource for ensuring this work is theoretically sound. I routinely pitched ideas to Anil and always received honest, valuable feedback.

I am grateful to have Ananth as an external committee member. Ananth's research addresses challenges in computational biology similar to those expressed in this thesis, and his work has inspired my own. I am thankful for Ananth making special arrangements to accommodate all of my committee meetings. He graciously set up a temporary office at the airport for my research defense; fortunately, he made it through security during the meeting.

I have been blessed with an overflowing fountain of support from my family. I express my deepest gratitude to my parents, Dennis Poirel and Sandra Stevens-Poirel, for providing a lifetime of love and encouragement. My parents stimulated my enthusiasm for academics when I was young and have been unquestioningly supportive of my decisions throughout my research career. They have always worked hard to ensure I received the best available education and experiences. My mother's determination to finish her PhD much later in life has provided me with a special source of inspiration. Thanks to my brother Tony for encouraging me in an unspoken way that only a big brother can. Thanks to Jim Hyatt for his unconditional support.

My grandparents, Weston "Posh" and Emma "Mos" Stevens, provided continual reminders of their love from back home. Thanks to Mos for routinely shipping boxes of

homemade goods overnight; the care packages you and Mom put together truly made me feel at home. Posh will forever hold a special place in my heart for his calm and unending words of reassurance.

I am indebted to the love of my life Brooke Cox for her support throughout graduate school. She attentively listened to my ramblings when courses, coding, and manuscript submissions simply did not proceed as planned. Spending time with Brooke and our pooches always offered a refreshing respite during the most stressful times. A unique thanks go to Rusty and Huck for their unremitting attention.

Thanks to my wonderful circle of friends in Blacksburg, many of whom participated in our weekly Drinking Club, which served as a revitalizing break from the long hours in Torgersen Hall. I have forged too many friendships through my studies in Blacksburg to explicitly name them all here. However, I would be remiss not thank Joseph Turner for our numerous enlightening discussions over coffee, beer, and board games. Thanks to Kirk Cameron for always offering your unabashedly honest advice and for the internship opportunity at MiserWare.

This work would not have been possible without contributions from an excellent supporting cast. Murali was intimately involved with every chapter. It was a pleasure working with Chris Lasher to write Chapter 2 on response networks. Conley Owens made significant initial progress on our algorithm for network-based functional enrichment (Chapter 3), and I enjoyed working with him to complete the project. Thanks to Ahsanur Rahman, Richard Rodrigues, Arjun Krishnan, and Jacqueline Addesa for their contributions to the (many) submissions and revisions of our work on network reconciliation (Chapter 4). Thanks to Richard Rodrigues, Jean Peccoud, David Ball, Neil Adames, John Tyson, Kathy Chen, and Pavel Kraikivski for the numerous suggestions during our weekly HGTV meetings that greatly improved LINKER and its application to the yeast cell cycle (Chapter 5). Thanks to Naveed Massjouni, David Badger, and Craig Estep for developing and maintaining GraphSpace. This work is unpublished but facilitated seamless collaboration with other research groups and expedited the analyses in Chapters 5 and 6. Thanks to Anna Ritz, Hyunju Kim, and Allison Tegge for their contributions to the work on automated reconstruction of human signaling pathways (Chapter 6). It was my pleasure to work with a visiting undergraduate, Nina Blanson, during the summer of 2013 on extensions to the shortest paths algorithm used by LINKER; Nina is a promising young researcher with a bright future. A special thanks go to past and present members of Murali's research group for their insightful feedback and constructive criticisms. I wish all of my collaborators the best in their future endeavors.

I am thankful for financial support provided by the Department of Computer Science Doctoral Fellowship and the National Science Foundation Graduate Research Fellowship Program.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation for this thesis

Functioning of the living cell is controlled by a complex network of interactions among genes, proteins, and other molecules. A major goal of systems biology is to understand and explain the mechanisms by which these complex interactions govern the cell's response to various conditions. Molecular interaction networks have proven to be a powerful representation for studying cellular behavior. Each node in these networks represents a gene or gene product (protein), and each edge indicates a molecular interaction between a pair of nodes (e.g., two proteins physically interacting in the cell). Depending on the type of the interactions and the experimental procedure used to identify the interactions, undirected or directed edges may be used to model interacting proteins. For example, a synthetic lethal genetic interaction identifies a pair of genes x and y whereby knocking out the activity of gene x or y alone results in a viable cell, but knocking out both genes simultaneously results in cell death. These interactions are often modeled as undirected edges. Conversely, a transcriptional regulatory interaction identifies a protein and a gene whereby the protein increases or decreases expression of the gene. Such interactions are naturally directed from the protein to the regulated gene. Graph-based representations of protein interactions provide a powerful scaffold for reasoning about cellular activity.

Advances of experimental techniques have led to an unprecedented production of -omics data available in public databases. We can combine multiple sources of data to construct an

organism's *interactome*, a network that includes the universe of experimentally-verified proteins and interactions in an organism. The interactome provides only a static representation of the cell and includes interactions that occur under a diverse ensemble of cellular conditions. Numerous algorithms have been developed to unravel the complexity of these networks. Our work addresses the drawbacks of existing techniques. *This thesis includes three related research efforts that introduce network-based approaches to bridge current methodological gaps in systems biology.*

i. Functional enrichment methods provide a summary of biological functions that are over-represented in an interesting collection of genes (e.g., highly differentially expressed genes between a diseased cell and a healthy cell). Standard functional enrichment algorithms ignore the known interactions among proteins. We propose a novel network-based approach to functional enrichment that explicitly accounts for these underlying molecular interactions. Through this work, we close the gap between set-based functional enrichment and topological analysis of molecular interaction networks.

ii. Many techniques have been developed to compute the response network of a cell. A recent trend in this area is to compute response networks of small size, with the rationale that only part of a pathway is often changed by disease and that interpreting small subnetworks is easier than interpreting larger ones. However, these methods may not uncover the spectrum of pathways perturbed in a particular experiment or disease. To avoid these difficulties, we propose to use algorithms that reconcile case-control DNA microarray data with a molecular interaction network by modifying per-gene differential expression p-values such that two genes connected by an interaction show similar changes in their gene expression values.

iii. Top-down analyses in systems biology can automatically find correlations among genes and proteins in large-scale datasets. However, it is often difficult to design experiments from these results. In contrast, bottom-up approaches painstakingly craft detailed models of cellular processes. However, developing the models is a manual process that can take many years. These approaches have largely been developed independently. We present LINKER, an efficient and automated data-driven method that analyzes molec-

ular interactomes. LINKER combines teleporting random walks and k-shortest path computations to discover connections from a set of source proteins to a set of target proteins. We demonstrate the efficacy of LINKER through two applications: proposing extensions to an existing model of cell cycle regulation in budding yeast and automated reconstruction of human signaling pathways. LINKER achieves superior precision and recall compared to state-of-the-art algorithms from the literature.

These research efforts address gaps in diverse areas of systems biology. Introducing and summarizing the relevant background information for each problem is not appropriate for a single chapter. Thus, we provide a thorough literature review for each application in the relevant chapters.

1.2 Organization and contributions of this thesis

Here, we describe the organization of the remaining chapters. This research is divided into two major groups: response networks and network connections. Chapters 2–4 introduce response networks, discuss their drawbacks, and provide alternative strategies for analyzing molecular interaction networks. Chapters 5 and 6 present two applications of our approach for connecting sets of proteins within a network.

Chapter 2. Chapter 2 surveys graph-theoretic approaches developed to compute biological *response networks*. A response network is a collection of active interactions upon exposing the cell to a certain condition or a particular stress. These methods typically integrate an organism's interactome with treatment-control gene expression data (see Section 2.1 for an explanation of this data). The differential expression value of a gene provides a proxy for its level of activity under the condition being studied. By weighting each node in the interactome with its corresponding gene's activity score, the algorithms surveyed in Chapter 2 identify subnetworks of the interactome whose nodes are collectively perturbed. Chapters 3 and 4 identify and propose solutions to the shortcomings of existing response network algorithms.

Chapter 3. Methods such as the response network algorithms discussed in Chapter 2 often yield subnetworks of the interactome containing hundreds or thousands of nodes and up to an order of magnitude more edges. Interpreting these large networks can be daunting; thus, it is desirable to summarize the biological information in such networks. A common approach is to use gene function enrichment analysis for this task. Functional enrichment analysis identifies biological functions that are overrepresented by the nodes in the given network. However, a major drawback of such methods is that they ignore information about the edges in the response network being analyzed. They treat the response network simply as a set of proteins and identify biological functions that annotate a statistically significant number of proteins in the set. In Chapter 3, we present a novel network-based method for functional enrichment that directly takes into account the pairwise relationships among the proteins. We show that our approach naturally generalizes Fisher's exact test, a widely-used set-based functional enrichment method. We demonstrate the utility of network-based enrichment through applications in three different organisms: human, rat, and baker's yeast.

Chapter 4. In Chapter 4, we discuss four approaches that reconcile treatment-control gene expression data with the interactome. The response network algorithms presented in Chapter 2 use a similar data integration to identify high-scoring subnetworks. However, the algorithms we discuss in Chapter 4 take a different approach; rather than identifying highly-relevant subnetworks, we compute a score for every node in the interactome, such that each node's final score is proportional to its relevance to the treatment being studied. These methods operate on the principle that two genes whose products physically interact should show similar changes in their gene expression values between treatment and control. We use the per-gene differential expression values as prior indication of each gene's relevance to the treatment. We then allow the expression values to change such that two genes whose products are connected by an interaction have similar values, but we ensure that these values do not deviate too much from their original settings. We enumerate four desirable properties that this class of algorithms should address. Using a compendium of gene expression data from 54 diverse human diseases, we comprehensively evaluate the extent to which each algorithm addresses these desired properties. We assess the final gene rankings given under

each disease to ensure the rankings maintain disease specificity, we investigate topological properties of top-ranking genes for each disease, and we perform functional enrichment tests on top-ranking genes. These reconciliation algorithms offer an attractive alternative to the response network algorithms discussed in Chapter 2.

Chapter 5. Chapter 5 describes our efforts to coalesce two common approaches for studying cellular behavior: top-down analysis of high-throughput interaction networks and bottom-up mechanistic models. Top-down methods build a comprehensive interactome with thousands of nodes (proteins) and hundreds of thousands of interactions. These methods subsequently mine the interactome for response networks (Chapter 2), functional subunits, or interesting motifs, among others. However, these approaches cannot simulate the dynamics of a specific biological system. Bottom-up models are typically built by careful analysis of hundreds of individual reactions in the literature. The parameters that govern the behavior of these models are then semi-manually tuned until the model accurately predicts experimental phenotypic behavior, e.g., the change in the growth of a yeast cell when a gene is mutated. One drawback to building bottom-up models is that the process is time-consuming and often involves a trial-and-error process to identify novel extensions and improvements of the model. We present top-down analytical methods that scour publicly-available interaction datasets to prioritize potential extensions to current bottom-up models, thereby expediting the modeling process. We apply our approach to a comprehensive interactome of *Saccharomyces cerevisiae* (yeast) and a well-known bottom-up model of the yeast cell cycle process [21].

Chapter 6. Cells are constantly exposed to external signals. Signaling pathways receive those signals at the cell's surface and propagate the signal through the cell and into the nucleus, ultimately eliciting a transcriptional response. Identifying and understanding cellular signaling pathways is a major challenge in systems biology. We focus on representations of signaling pathways that begin at a set of signaling receptors and terminate at a collection of downstream transcriptional regulators and transcription factors. Currently, manually-curated signaling pathways represent the collective knowledge of cellular signaling.

We convert available signaling pathways to appropriate graph representations and apply the concepts from Chapter 5 to these pathways. We demonstrate that our approach can faithfully reconstruct pathway representations from existing databases.

Relationships to research projects. Chapter 2 surveys several response network algorithms that are referenced throughout the thesis. These algorithms represent the types of top-down network-based algorithms discussed in the remaining chapters. Chapter 3 directly addresses our first research effort of moving from set-based to net-based functional enrichment analysis. Chapter 4 highlights our progress on the second research effort to bridge the gap between transcriptomics and the interactome. Chapters 5 and 6 target our final research effort to bridge the gap between top-down and bottom-up methodologies. The methods presented in Chapters 4–6 incorporate teleporting random walks. These chapters highlight a variety of systems biology applications of random walks on the interactome.

Chapter 2

Response Networks

Christopher D. Lasher, **Christopher L. Poirel**, and T. M. Murali. Cellular response networks. In *Problem Solving Handbook in Computational Biology and Bioinformatics*, pages 233-252, editors Lenwood S. Heath and Naren Ramakrishnan. Springer US, 2011.

2.1 History of the problem

Complex networks of interactions between genes, proteins, and other molecules choreograph cellular processes. The interactions that are active in the cell change over time, both as a natural outcome of the cell's life cycle and in response to external signals. The set of active interactions, called the *response network*, are likely to be significantly different between a normally-functioning cell and a diseased cell. The wide availability of DNA microarray data and experimentally-determined interaction networks has made it possible to automatically compute response networks. This chapter surveys algorithms that have been developed to compute response networks.

Genes carry genetic information that is used to synthesize essential components of the living cell. These components are called gene products, typically RNA molecules or proteins. Coordinated interactions among gene products comprise and control many fundamental cellular processes such as the formation of protein complexes, the metabolism of food by biochemical pathways, and signaling pathways triggered by external signals. Gene products

also control and modulate the synthesis and activity of other gene products. These interactions constitute an intricate network that dynamically changes in response to a myriad of cues. Therefore, discovering *response networks*, the set of molecular interactions that are active in a given cellular context, and understanding how normal response networks may be perturbed in a disease are fundamental biological questions [51].

Gene expression is the process by which a gene is first transcribed to messenger RNA (mRNA). The *expression level* of a gene is the number of copies of its mRNA that are present in a cell. DNA microarrays have allowed biologists to simultaneously measure the average expression level of each gene in a set of cells. DNA microarrays offer a powerful experimental platform to study diverse contexts, since they capture a snapshot of the activity of all genes in the cells in the sample. However, DNA microarrays measure levels of the *nodes* (genes) and do not directly provide any information on the *edges* (interactions). Data regarding edges are available from datasets of physical and functional interactions between genes and proteins that are now widely available. Integrated analysis of gene expression data and protein-protein interaction (PPI) networks is emerging as a powerful technique for computing response networks. This chapter surveys several algorithms that are available to perform this type of analysis.

This type of analysis is distinct from methods that find modules in PPI networks alone. Such analysis is usually performed on protein interaction networks integrated from a variety of different experimental sources and public repositories. However, an experiment that reports an interaction often does not yield information on the conditions under which that interaction takes place in the cell. In many situations, the experimental context in which an interaction happens is lost when the interaction is recorded in a database. In other cases, the context may simply not be apparent. For instance, an interaction between two human proteins may be detected by a yeast 2-hybrid experiment [34]. Since such an experiment is performed in *Saccharomyces cerevisiae* (baker's yeast), it simply cannot produce any information on when the detected interaction may take place in a human cell. As a consequence, protein interaction networks typically represent the *universe* of interactions that take place in multiple, different contexts within the cell. Integrating them with measurements of molecular levels, such as DNA microarray data, is necessary for computing response networks.

2.2 Detecting response networks

We divide response network algorithms into two broad classes, depending on the design of the experiment used to collect DNA microarray data:

1. A very common experimental design partitions the set of samples into two subsets, with one subset corresponding to an experimental treatment and another subset corresponding to a control. Numerous methods have been developed to assess to what degree each gene is differentially expressed when comparing the treatment to the control. Using a hypothesis testing framework, for each gene g, these methods yield a p-value $0 \leq p_g \leq 1$ representing the statistical significance of the difference between the two sets of expression levels of the gene. These p-values form the starting point of response network computations. We call such datasets *treatment-control* data and examine these methods in Section 2.2.1.

2. Another common experimental design yields a gene expression dataset consisting of measurements from multiple samples under a particular experimental condition; the samples can correspond to multiple time-points after exposing cells to a particular treatment or stimulus or to multiple patients diagnosed with a particular disease. The complete gene expression data is part of the input to an algorithm to compute response networks. Analysis of such datasets usually starts by computing co-expression or similarity values for gene pairs. We discuss *co-expression-based* techniques in Section 2.2.2.

2.2.1 Detecting response networks from treatment-control data

Experiments for analyzing gene expression often produce treatment-control data. The treatment samples offer measurements of the expression of different genes under a certain experimental condition or phenotype (e.g., after a gene knock-out or for a specific disease). The control samples measure gene expression without the influence of the experimental condition (e.g., wild-type cells or normal cells).

We discuss three algorithms that integrate this type of gene expression data with molecular interaction networks. The ActiveModules algorithm of Ideker *et al.* [50] and the

algorithm of Dittrich *et al.* [30] estimate the differential expression of each node in the protein-protein interaction network and subsequently find high-scoring subnetworks, i.e., subgraphs that have large differential expression in total. The DEGAS algorithm of Ulitsky and Shamir [127] uses a different approach: for each gene, the method computes a separate p-value in every sample in the treatment. After combining this expression data with a protein-protein interaction network, the algorithm searches for a minimally connected subnetwork of genes that respond to the experimental condition for at least some specified number of samples in the treatment.

The inputs to the algorithms discussed in this section are an undirected protein-protein interaction network $G = (V, E)$ and two sets of gene expression data, $V_T = \{g_T \mid g \in V\}$ and $V_C = \{g_C \mid g \in V\}$. Here, T is the set of samples in the treatment and $g_T : T \to \mathbb{R}$ denotes the expression values of gene g in each of the samples in T, and C is the set of samples in the control and we define g_C analogously to g_T. Informally, the goal of these methods is to compute the connected subgraph of G such that the genes in the subgraph show the most differential expression between the samples in T and the samples in C.

The ActiveModules algorithm

Ideker *et al.* [50] introduce the ActiveModules algorithm for computing highly-perturbed response networks from treatment-control data. For each gene g in G, they compute a p-value p_g based on the expression values g_T and g_C of that gene in the treatment samples and the control samples. Many tools are available to calculate such p-values [33]. For instance, a simple approach is to apply the t-test to g_T and g_C. In general, the p-value represents the statistical significance of the observed difference between the expression levels of a gene in T and in C. A smaller p-value indicates a more statistically significant difference. Ideker *et al.* [50] convert each value p_g to a z-score z_g using the inverse normal cumulative distribution function evaluated at $1 - p_g$, i.e., $z_g = \Phi^{-1}(1 - p_g)$, where Φ is the cumulative normal distribution function. This transformation converts small p-values to large z-scores. Consequently, connected subnetworks composed of genes with high z-scores are desirable.

The authors do not simply discard genes with low z-scores. Instead, they develop a method for scoring any subgraph of G based on the z-scores of all the genes in the subgraph.

For a subgraph A of G on a k-node set B, define the Liptak-Stouffer z-score z_A as

$$z_A = \frac{\sum_{g \in B} z_g}{\sqrt{k}}.$$

Clearly if A is a subgraph of genes with high z-scores, then it will have a large aggregate Liptak-Stouffer z-score and may possess some biological significance. The final step in scoring a subgraph lies in determining whether or not z_A is statistically significant. Ideker *et al.* [50] compute the statistical significance empirically: they compute the aggregate Liptak-Stouffer z-scores for multiple subgraphs induced by k randomly selected genes, and estimate the mean μ_k and standard deviation σ_k of these random subgraphs of size k. They define the *corrected subgraph score* s_A as follows,

$$s_A = \frac{z_A - \mu_k}{\sigma_k}.$$

This transformation adjusts the z-score z_A so that a randomly-selected subgraph on k nodes will have a corrected subgraph score with mean 0 and standard deviation 1.

With a function to score subgraphs in hand, the authors proceed to discover high-scoring subgraphs. They demonstrate that a similar problem is NP-complete (we describe this problem in more detail in Section 2.2.1, page 13). Thus, it is unlikely that an efficient (polynomial time) algorithm exists that computes the subgraph that maximizes s_A. Ideker *et al.* [50] resort to simulated annealing [67], a heuristic method often used to solve computationally intractable combinatorial optimization problems. The following algorithm demonstrates the simulated annealing technique. The algorithm has three user-determined parameters: n, the number of iterations, a starting temperature T_s and an ending temperature $T_e < T_s$.

The variable T represents a temperature that decreases geometrically with each iteration, by a factor of $(T_e/T_s)^{1/n}$. The algorithm always accepts a modification that increases the corrected subgraph score. However, when $s_I < s$, the algorithm accepts the change with a probability $0 < p = e^{(s_I - s)/T} < 1$. For a fixed value of T, the closer s_I is to s, the closer p is to 1. For a fixed value of $s_I - s$, the probability p decreases as T decreases, indicating that the algorithm is more liberal in earlier iterations, being more likely to keep changes that lower the corrected score. Since the returned graph induced by I is not guaranteed to be

Algorithm 2.1 Simulated annealing algorithm to compute high-scoring subgraphs.

SimAnneal$(G(V,E), n, T_s, T_e)$

1: Label each node in V either *in* or *out* with equal probability, and let I be the set of all nodes labeled *in*
2: Compute s_I
3: $T \leftarrow T_s$
4: **for** $i = 1 \ldots n$ **do**
5: $s \leftarrow s_I$
6: Select a node $v \in V$ uniformly at random and switch its label
7: Compute s_I
8: **if** $s_I > s$ **then**
9: Keep the new label for v
10: **else**
11: Keep the new label for v with probability $e^{(s_I - s)/T}$
12: $T \leftarrow T \times (\frac{T_e}{T_s})^{\frac{1}{n}}$
13: **return** The subgraph of G induced by I

connected, the authors simply take the highest-scoring connected component as the result. Note that this approach will not necessarily find the optimal solution, but operates under the belief that any high-scoring subnetwork likely provides some biological insights.

The algorithm of Dittrich *et al.*

Dittrich *et al.* [30] build on the Ideker *et al.* [50] approach by developing a new scoring function and a different method for discovering high-scoring subgraphs. First, they follow Pounds and Morris [100] to model the distribution of p-values over all genes in V as a mixture of noise and signal components. Let $B(a,b)$ denote the beta distribution, where a and b are the two parameters that define the shape of the beta distribution function. The probability density function of $B(a,b)$ is

$$f(x) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1}(1-x)^{b-1},$$

where $\Gamma(x) = \int_0^\infty t^{x-1}e^{-x}dt$ is the gamma function. Dittrich *et al.* assume that the signal component of the distribution of p-values has a $B(a,1)$ distribution, i.e., given that a p-value x is generated by the signal component, its probability distribution function is ax^{a-1}. Similarly, they assume that if a p-value is generated by the noise component, then the p-value is $B(1,1)$

or uniformly distributed on (0,1). Therefore, if λ (respectively, $1-\lambda$) is the probability that a p-value is generated by the noise (respectively, signal) component of the mixture, then the probability distribution function for a p-value x can be rewritten as

$$f(x|a,\lambda) = \lambda + (1-\lambda)ax^{a-1}, \qquad 0 < x \leq 1; 0 < \lambda < 1,$$

where λ and a are mixture and shape parameters, respectively. Given the p-values $P_V = \{p_v | v \in V\}$, define the likelihood of these values as

$$\mathcal{L}(\lambda, a; P_v) = \prod_{v \in V} \left(\lambda + (1-\lambda)ap_v^{a-1}\right).$$

The authors use numerical optimization methods to estimate the mixture and shape parameters $[\lambda^*, a^*] = \arg\max_{\lambda,a} \mathcal{L}(\lambda, a; P_V)$ that maximize the likelihood of the p-values.

The ultimate goal of this approach is to develop a scoring function that associates a p-value arising primarily from the signal component with a positive score and a p-value generated by background noise with a negative score. The following scoring function captures this property:

$$s_x = \log \frac{ax^{a-1}}{a\tau^{a-1}} = (a-1)\left(\log(x) - \log(\tau)\right),$$

where τ is a p-value threshold that yields a user-specified false discovery rate [1] (see Pounds and Morris [100] for details). A p-value is deemed significant when it is smaller than τ, thus the corresponding node is assigned a positive score. Conversely, if a p-value is larger than τ, the corresponding node is assigned a negative score. Dittrich *et al.* define the score s_A for a subgraph A as

$$s_A = \sum_{g \in A} s_{p_g},$$

that is, s_A is simply the sum of the scores for each of the nodes in A. Dittrich *et al.* aim to compute the subgraph of G with the largest score. This problem is known as the *maximum-weight connected subgraph* (MWCS) problem: given a graph $G = (V, E)$ and node weight $w_v \in \mathbb{R}$ for each $v \in V$, the MWCS problem asks for the connected subgraph $G' = (V', E')$

[1] The false discovery rate is the ratio of false positives (values incorrectly identified to be significant) and the total number of values deemed to be significant.

of G that maximizes $w_{G'} = \sum_{v \in V'} w_v$. The MWCS problem has been proven to be NP-complete [50]. Notice that the MWCS problem is trivial if all weights are positive, since the entire graph G would clearly be the optimal solution.

Dittrich *et al.* convert an instance of the MWCS problem into an instance of the prize-collecting Steiner tree (PCST) problem. While the PCST problem is also NP-complete, Ljubić *et al.* [75] provide an elegant algorithm based on formulating the PCST problem as an integer linear program (ILP). They propose a branch-and-cut heuristic to solve this ILP. The algorithm does not have a running time that is polynomial in the size of the input. However, Dittrich *et al.* show that this approach finds provably-optimal solutions in a reasonable amount of time for biologically-relevant network sizes.

The DEGAS algorithm

Ulitsky *et al.* [127] develop the DEGAS algorithm for identifying disease-related pathways within the cell. In contrast to the algorithms discussed previously, Ulitsky *et al.* calculate multiple p-values for each gene, one for each sample in T. The main goal of the algorithm is to discover subgraphs containing several genes that are differentially expressed in multiple samples in T. The authors call such subgraphs *dysregulated pathways* (DPs). The process is two-fold. First, discover minimal connected subgraphs that have at least k differentially-expressed genes, where k is a parameter to the algorithm. Second, they find those minimal connected subgraphs that are statistically significant.

Recall that T is the set of n treatment samples. Associate with each node $v \in V$ a set of treatment samples $S_v \subseteq T$ in which v is differentially expressed (in comparison to the expression of v in the control samples C). For every node $v \in V$, Ulitsky and Shamir compute the set S_v by (i) estimating a p-value in each sample $t \in T$ that represents the differential expression of g in t, (ii) applying a user-specified cutoff on the p-values, and (iii) including a treatment sample t in S_v if the p-value is below the cutoff. The authors construct a bipartite graph $B = (V, T, E^B)$, where $E^B = \{(t, v) | t \in S_v\}$. The graph B is simply a bipartition between genes and samples, with an edge between gene v and sample t when $t \in S_v$. Now define a subset $C \subseteq V$ of genes to be a *connected* (k, l)*-cover* $CC(k, l)$ if the following two conditions hold:

1. C induces a connected subgraph in G.

2. There exists a set of $n-l$ treatment samples $T' \subseteq T$ such that for every sample $t' \in T'$, $|N(t') \cap C| \geq k$, where $N(t')$ is the set of genes that are adjacent to t' in the graph B.

The second property of a connected (k,l)-cover C states at least k genes in C are differentially expressed in all but l samples in T. This notion ties together a connected subgraph of G with a set of treatment samples, in each of which a sufficiently large number of genes in the subgraph are perturbed. By not requiring all genes in C to be perturbed in all samples in T', a connected (k,l)-cover is able to accommodate inter-sample variation and experimental noise.

Given integers k and l, the *minimum connected* (k,l)*-cover* problem $MCC(k,l)$ is to find the connected (k,l)-cover with the fewest number of nodes. Since discovering minimal connected subgraphs is NP-hard, Ulitsky *et al.* develop approaches that offer provably good results. They propose Covering Using Shortest Paths (CUSP), an algorithm which provides a $k(n-l)$-approximation for $MCC(k,l)$. Define the *distance* between two nodes $d(u,v)$ as the minimum number of edges in any path connecting u and v in G. The algorithm proceeds in four major steps:

1. Find the k shortest paths from each node $r \in V$ to each sample $u \in T$. More specifically, for each node $r \in V$, for each sample $u \in T$, and for each $1 \leq i \leq k$, let $P[r,u]_i$ be the ith closest node to r in G that is a neighbor of u, and let $D[r,u]_i = d(r, P[r,u]_i)$. Compute $D[r,u]_i$ and $P[r,u]_i$ for $1 \leq i \leq k$.

2. Find a set of $n-l$ samples in T for which the k shortest paths from r are not very long. Specifically, compute S_r, the set of $n-l$ samples in T that have the smallest values for $m[r,u] = max_q\{D[r,u]_q, 1 \leq q \leq k\}$. In other words, compute $m[r,u]$ for each sample $u \in T$ and include the $n-l$ samples that have the smallest values of $m[r,u]$ in the set S_r.

3. Extract the shortest paths between r and the samples in S_r, i.e., compute X_r, the union of the paths to the nodes in G that neighbor the samples in S_r. The authors claim that X_r is a $CC(k,l)$ in G. Indeed, it induces a connected component in G.

Furthermore, each of the $n - l$ samples in $S_r \subseteq T$ is covered once for each $P[r, u]_i$ where $1 \leq i \leq k$.

4. Output the smallest X_r, i.e., return the $CC(k, l)$ instance $X = \arg\min_{v \in V} |X_v|$, which is designated to be a DP.

To assess the statistical significance of the DP returned by the CUSP algorithm, the authors generate multiple random networks with the same number of nodes as G, using degree preserving randomization [83]. They apply the CUSP algorithm on the original graph B with different values of k. They also run the CUSP algorithm on each random network with different values of k, and compute a distribution of DP sizes for each value of k. The p-value of each DP computed in B is the fraction of DPs with a larger size computed in the random graphs. They return the most statistically-significant DP computed from the original network that corresponds to this k.

Notice that the CUSP algorithm computes only one DP. In practice, we want to return multiple DPs and test each of them for statistical significance, since any significant DP may be biologically interesting. The authors describe a method for discovering multiple DPs. Suppose X is the first DP returned by the CUSP algorithm. For each node $v \in X$ remove all edges adjacent to v from E^B in the graph B and call the resulting graph B'. Then run CUSP on B' to produce a new DP. Continue this procedure until CUSP no longer returns a statistically significant DP.

2.2.2 Co-expression-based methods to detect response networks

Gene expression data sets with many samples per condition or phenotype, or with samples from many conditions or phenotypes, afford the opportunity to calculate co-expression, or similarity values for gene pairs. These datasets give rise to another class of algorithms that use similarity values to detect response networks. We discuss four different approaches to integrating expression and network data in order to determine response networks. The first approach, by Hanisch *et al.* [44], simultaneously uses gene expression similarity and distances in the PPI network and clusters genes after combining these distances. The second approach, by Murali and Rivera [87], overlays expression-based similarities on the edges of the PPI

network as edge weights, and then detects heavy subgraphs in the weighted network. The final pair of closely-related approaches, by Ulitsky and Shamir [126, 128], indirectly leverage the PPI network to constrain the actions for refining sets of similarly expressed genes.

The algorithms discussed in this section use as inputs one undirected protein-protein or protein-reaction interaction network $G = (V, E)$ and one gene expression dataset $V_S = \{g_S \mid g \in V\}$, where S is the set of samples and $g_S : S \to \mathbb{R}$ denotes the expression values of gene g in the samples S. Informally, these methods strive to compute a connected subgraph of G such that the genes in the subgraph show the most similar expression across the samples in S.

The algorithm of Hanisch *et al.*

Hanisch *et al.* [44] present an algorithm that clusters genes using distances between their expression profiles in combination with distances between their gene products in a PPI network. They begin by converting curated metabolic pathways from KEGG [58] into a bipartite graph $G = (V, E)$ with biological molecules (e.g., enzymes and metabolites) as one set of nodes and reactions as the other set. Edges in G connect molecules to reactions in which they participate. In order to emphasize the relationships between the biological molecules and to disfavor paths through ubiquitous molecules which take part in many reactions (e.g., ATP), each edge e receives a weight w_e equal to the degree of the incident biological molecule. The distance $d_{\text{net}}(u, v)$ between two nodes u and v in this bipartite graph is the cost of the shortest path between u and v. Hanisch *et al.* then calculate the distance in expression $d_{\text{exp}}(g, h)$ for all pairs of genes g and h, as $1 - c(g, h)$, where $c(g, h)$ is the Pearson's correlation coefficient between g_S and h_S.

After mapping genes in the gene expression data to the enzymes they code for in G, the authors combine the computed distances d_{exp} and d_{net} into a joint distance $\Delta(u, v)$ as follows:

$$\Delta(u, v) = 1 - \frac{\lambda_{\text{exp}}(u, v) + \lambda_{\text{net}}(u, v)}{2},$$

where the *logistic regression* function $\lambda_\Psi(u, v)$, $\Psi \in \{\text{exp, net}\}$, is

$$\lambda_\Psi(u, v) = \frac{1}{1 + e^{-s_\Phi(\delta_\Phi(u,v)-\nu_\Phi)}}.$$

User defined parameters s_Φ and ν_Φ control the shape of the logistic curve, giving the slope of the curve and the point at which the curve reaches $\frac{1}{2}$, respectively. Hanisch *et al.* set the values of these parameters empirically.

Finally, Hanisch *et al.* use agglomerative hierarchical clustering to partition the genes into a user-defined number of groups. This aspect distinguishes their algorithm from the others presented below, which do not require pre-defining the number of computed response networks. To assist with choosing an appropriate number of clusters, Hanisch *et al.* plot silhouette values [105], which measure the separation and tightness of clusters, for different cut points, and heuristically select appropriate points. As a final observation, although this algorithm does not directly compute response networks, by taking distances in the bipartite graph into account, it indirectly discovers those metabolic pathways perturbed in an experiment.

The ActiveNetworks algorithm

The ActiveNetworks algorithm presented by Murali and Rivera [87] projects co-expression values as edge weights onto an interaction network and casts the problem of finding response networks as one of finding dense subgraphs within the weighted interaction network. First, they remove all genes with little variation in expression and their incident edges from G. Next, they compute the weight w_e of each edge $e = (g, h)$ in E as the absolute value of Pearson's correlation coefficient of g_S and h_S. Murali and Rivera then assess the statistical significance of the weight of each edge in the PPI network using a permutation test and remove edges with insignificant weights from G.

Given a subgraph $H = (V', E')$ of G, they define its *density* as

$$w_H = \frac{\sum_{e \in E'} w_e}{|V'|},$$

i.e., the total weight of the edges in H divided by the number of nodes in H. Computing the subgraph of maximum density can be solved in polynomial time [35] or by using linear programming [19]. In practice, Murali and Rivera use a greedy algorithm that guarantees a 2-approximation, i.e., the subgraph computed by the algorithm has density at least half as much as that of the most dense subgraph in G. Define the weight of a node to be the total weight of the edges incident on it. The algorithm repeatedly deletes the node of smallest weight until G is empty. It reports the most dense subgraph encountered during this process. Murali and Rivera embed this algorithm in a heuristic to find all "dense pockets" in G: apply the greedy algorithm to G, delete the edges of the computed subgraph from G, and repeat this process, until the density of G falls below its initial density. They return the union of all dense subgraphs computed as the response network. See Section 2.2.3 for an application of this method to data for *S. cerevisiae*.

The MATISSE algorithm

Ulitsky and Shamir [126] present an algorithm called MATISSE that seeks to find sets of genes (called *modules*) with high expression similarity, but with the additional constraints that (i) each set must induce a connected subgraph in the interaction network, and (ii) no gene appears in more than one set. In the context of this chapter, we consider the union of these modules to comprise a response network.

Ulitsky and Shamir begin by computing a likelihood ratio for each pair of genes from their similarity (measured as the value of Pearson's correlation coefficient): this likelihood ratio compares the probability that such similarity would be observed under the assumption that the two genes respond to the experiment versus the assumption the two genes have no relation. Large positive values of the logarithm of the likelihood ratio indicate greater support for the hypothesis that the two genes have related expression patterns and respond to the experimental condition. Conversely, large negative values of the logarithm indicate greater support that the two genes have unrelated expression patterns.

Ulitsky and Shamir then construct a complete similarity graph $X = (V, E, w)$, where the set V of nodes is the set of all genes, the set E of edges consists of all pairs of genes, and $w : E \to \mathbb{R}$ is a function specifying the log-likelihood for every edge in E. Given a set V' of

genes, they define the score $s_{V'}$ of this set of genes as the sum of the log-likelihoods of all pairs of genes in V'; they define the score for a set of gene sets as the sum of the scores for all gene sets in the set. They address the problem of finding multiple disjoint gene sets in X such that each set of genes induces a connected subgraph in the interaction network G, and the total score of the gene sets is as large as possible. The MATISSE algorithm finds these subgraphs in three stages: identification of small subgraph "seeds", improving subgraphs from the seeds, and, finally, identifying statistically significant subgraphs.

To detect seeds, Ulitsky and Shamir settle on a "best-neighbors" heuristic that operates as follows. First, rank all nodes in X by the sum of their edge weights. Next, take the subgraph induced by the top ranked node and all the nodes connected to it in X by edges with positive weight as a seed. Remove this subgraph from X. Repeat the process with the next remaining highest-ranked node until X is empty.

They proceed to simultaneously refine all seeds using a greedy algorithm. At each step, they add a node to a module, remove a node from a module, reassign a node from one module to another, or merge two modules. They proceed with an action if it increases the overall score and maintains the connectivity of the subgraph induced by each module in G. This procedure terminates when no action meets these criteria.

In the final stage, the algorithm reports modules that are statistically significant. The authors use the following approach that is standard in the literature. Given a module, sample sets of genes of the same size from X, and compute the score of each set of genes. Next, compute the rank of the module's score among the scores of these random gene sets, and set the statistical significance of the module to be its score's rank divided by the number of sampled gene sets.

The CEZANNE algorithm

In the CEZANNE algorithm, Ulitsky and Shamir further extend MATISSE to accommodate the situation when each edge in G has a weight that indicates the probability that it is a true interaction in the cell [128]. They restate their objective as one of finding disjoint modules of strongly co-expressed genes in X that have a high probability of connectedness in G. More formally, given a user-specified probability q, the algorithm detects subsets of genes

that induce node-disjoint modules in X that are *q-connected* in G, i.e., have a probability of connectedness of at least q.

For each edge $e \in E$, let p_e denote the probability that e is a true interaction. The authors assign a confidence value $-\log(1-p_e)$ to the edge e. Let G_U be the subgraph of G induced by the set of genes $U \subseteq V$. Consider any *cut* of G_U, i.e., a set of edges in G_U that partition U into two non-empty subsets. The *weight* of this cut is the sum of the confidence values of the edges in this cut. With these definitions, a subset U of genes is *q-connected* if, for each possible cut in G_U, the weight of the cut is at least $-\log(1-q)$. (Note that $\log(1-q)$ represents the probability the subgraph is not connected.) Formally, a subset U is q-connected if, for all $W \subset U$

$$\sum_{e=(x,y),x\in W,y\in U-W} -\log(1-p_e) \geq -\log(1-q).$$

To determine if a subset is q-connected, it is sufficient to check if the weight of the minimum cut in G_U exceeds $-\log(1-q)$.

The steps for identifying disjoint modules that are q-connected closely follow those of MATISSE: seed identification, module optimization, and filtering for significant modules. CEZANNE starts with the modules computed by MATISSE; recall that MATISSE does not take edge weights into account. The authors identify q-connected seeds by recursively splitting the modules into smaller subgraphs along the minimum cut, until the weight of the cut is at least $-\log(1-q)$. The computations required to refine these seeds must satisfy the constraint that any modification must preserve q-connectedness (as opposed to connectivity). Ulitsky and Shamir employ several heuristics for performing the optimizations within acceptable running times; we refer the reader to their paper for details. Finally, Ulitsky and Shamir report only statistically-significant modules as follows: they create an empirical distribution of 100 scores by shuffling each gene's expression values among the samples, applying the CEZANNE algorithm, and recording the highest similarity score for each run. They assess the p-value for a module by ranking it within this distribution of similarity scores for randomized gene expression data.

2.2.3 Application of two co-expression based methods

We showcase the application of two co-expression-based methods for computing response networks to different stresses applied to *S. cerevisiae*.

Application of ActiveNetworks to amino acid starvation.

Murali and Rivera (unpublished) apply the ActiveNetworks algorithm to a time-course of DNA microarray data collected upon amino acid starvation [37] and an interaction network integrated by Kelley and Ideker from multiple sources [62]. This network contains 15,429 protein-protein interactions from the Database of Interacting Proteins (DIP) [108], 5869 protein-DNA interactions (between transcription factors and their target genes) [71], and 6306 metabolic interactions (interaction between proteins that operate on at least one common metabolite) based on the KEGG database [58]. As a negative control, this network includes 4812 genetic interactions [123]. Since genetically interacting genes are unlikely to be co-expressed, such interactions should not appear in a response network. Overall, this network contains 32,416 (27,604 physical and 4812 genetic) interactions among 5601 proteins.

Figure 2.1 displays a layout of the computed response network. At the center of this network are two transcription factors PHD1 and GCN4. PHD1 is a transcriptional activator that enhances pseudohyphal growth, a pattern of cell growth that occurs in conditions of nitrogen limitation and an abundant fermentable carbon source. GNC4 is a transcriptional activator of amino acid biosynthetic genes in response to amino acid starvation. The Gene Ontology (GO) [6] biological processes enriched in this network include purine ribonucleoside salvage, electron transport, glucose catabolism, carboxylic acid metabolism and gluconeogenesis, pointing to the intricate network of transcriptional regulatory interactions, protein complexes, signaling circuits, and metabolic pathways activated in response to the stress. The response network includes only two genetic interactions (not displayed in Figure 2.1), indicating that genetically interacting gene pairs are not highly co-expressed in this gene expression data set.

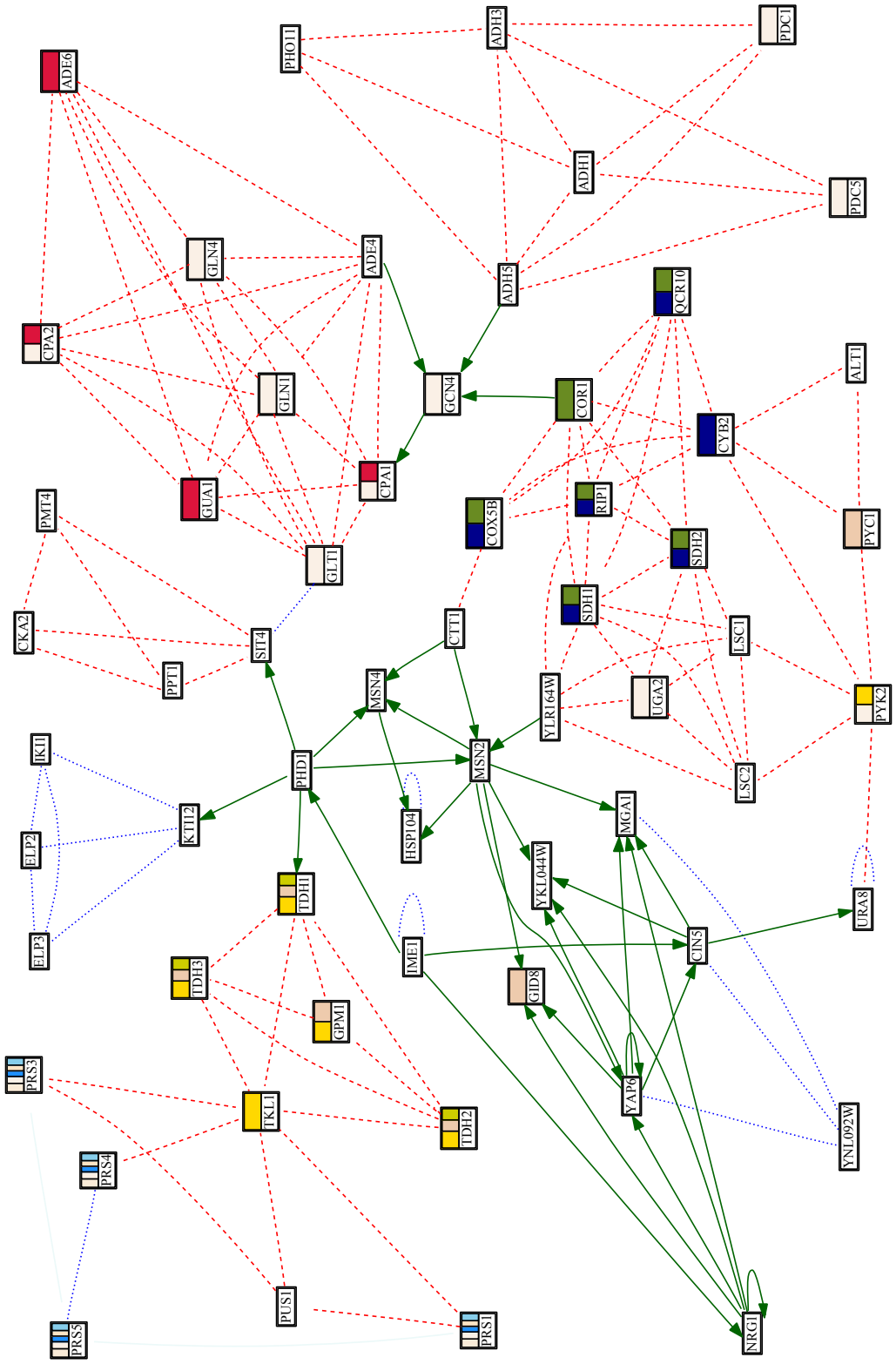


Figure 2.1: Response networks in *S. cerevisiae* following amino acid starvation showing interactions between transcription factors and their target genes (solid green arrows), PPIs (dotted blue), and metabolite-based interactions (dashed red).

Application of CEZANNE to DNA damage response.

To detect response networks of *Saccharomyces cerevisiae* (yeast) under conditions inducing DNA damage, Ulitsky and Shamir obtained expression data from Gasch *et al.* [36] and a PPI network with confidence values derived from previous work by Collins *et al.* [24]. Ulitsky and Shamir applied and compared the CEZANNE [128], MATISSE [126], and Hanisch co-clustering [44] algorithms, as well as methods that used only expression data and not the interaction network.

CEZANNE identified a total of 14 significant response networks covering 471 of the 6167 genes in the interaction network, ranging in size from 3 to 346 genes. All response networks were enriched for at least one term in the "biological process" category, and 11 were enriched in at least one term in the "molecular function" category. The largest response network featured many proteins associated with ribosomal biosynthesis; as a whole, the module experienced down-regulation in response to DNA damage. The other modules include enrichment for genes with annotations related to function in the ribosome, proteasome, and mitochondrion. CEZANNE was able to detect both modules not detected by the other methods as well as more specific and precise modules, as assessed by F-tests. Such modules included genes correlated in the literature to cell response to stress but not previously detected by the other methods.

2.3 Comparing response networks

At times, researchers wish to identify similarities and differences between multiple response networks. These approaches are motivated by the desire to compare the cell's response to different conditions. Noting that Ideker *et al.* [50] extend their ActiveModules algorithm (Section 2.2.1) to incorporate expression profiles for multiple experimental conditions, we discuss two other algorithms capable of handling multiple response networks, namely, Gene Network Enrichment Analysis (GNEA) and Network Legos. The increasing availability of public repositories containing thousands of gene expression datasets (e.g., the National Center for Biotechnology Information's Gene Expression Omnibus) encourages future development of novel methods for analyzing and comparing response networks.

2.3.1 Gene Network Enrichment Analysis

Motivated by the question of whether genes responsible for insulin production and uptake appeared frequently in response networks computed from contrasts of diabetic and non-diabetic patients, Liu *et al.* [74] develop a method for detecting enrichment of gene sets across a collection of response networks, which they call *gene network enrichment analysis* (GNEA). For each gene set F, they compute the significance of its enrichment in each response network using the one-sided version of Fisher's exact test and tally the number of response networks c_F for which the gene set has a statistically significant enrichment (p-value at most some user-defined threshold). To empirically determine the significance of c_F, Liu *et al.* construct a distribution of counts from 10,000 random gene sets of the same size as F: the p-value of c_F is the fraction of random gene sets whose counts are larger than c_F. Finally, they report all gene sets with highly significant counts.

2.3.2 Network Legos

Murali and Rivera [87] introduce the concept of *network legos* as a means for explicitly representing similarities and differences between response networks. They treat a response network simply as a set of edges. Given a collection $\mathcal{A}$ of response networks, each of which is a subgraph of an undirected interaction network G, they first define the notion of a *block* as a triple $(H, \mathcal{P}, \mathcal{N})$, where H is a subgraph of G, $\mathcal{P}$ and $\mathcal{N}$ are disjoint subsets of $\mathcal{A}$, and $\mathcal{P} \neq \emptyset$ such that

$$H = \Big(\bigcap_{P \in \mathcal{P}} P \Big) \bigcap \Big(\bigcap_{N \in \mathcal{N}} (G - N) \Big),$$

where "$\cap$," "$-$," and "$\cup$" respectively denote the intersection, difference, and union of the edge sets of two graphs and

1. $\mathcal{P}$ is maximal, i.e., there is no response network $P \in \mathcal{A} - \mathcal{P}$ such that $H \subseteq P$, and
2. $\mathcal{N}$ is maximal, i.e., there is no response network $N \in \mathcal{A} - \mathcal{N}$ such that $H \cap N = \emptyset$.

In other words, they form H by taking the intersection of all the response networks in $\mathcal{P}$ and removing any edge that appears in any of the response networks in $\mathcal{N}$. Informally, H represents the cellular response that is common to all the experimental conditions whose

response networks are members of the "positive" set $\mathcal{P}$. It also does not incorporate any aspect of the cellular response captured in the "negative" set $\mathcal{N}$.

Murali and Rivera reduce the problem of computing all blocks to that of computing all closed biclusters in an appropriately defined binary matrix representing presence and absence of interactions in each response network. They use the CHARM algorithm for this purpose [145]. To estimate σ_H, the statistical significance of an observed block $(H, \mathcal{P}, \mathcal{N})$, Murali and Rivera construct a set of blocks $R_H = \{(H', \mathcal{P}', \mathcal{N}')\}$, composed from random selections of response networks $\mathcal{P}'$ and $\mathcal{N}'$, where $|\mathcal{P}| = |\mathcal{P}'|$ and $|\mathcal{N}| = |\mathcal{N}'|$. They set σ_H to be the fraction of blocks in R_H whose subgraph has at least as many interactions as H.

Next, Murali and Rivera define a natural partial order between blocks: Given two distinct blocks $(H_1, \mathcal{P}_1, \mathcal{N}_1)$ and $(H_2, \mathcal{P}_2, \mathcal{N}_2)$, they say that $H_1 \prec H_2$ if

1. $\mathcal{P}_1 \subseteq \mathcal{P}_2$ and $\mathcal{N}_1 \subseteq \mathcal{N}_2$ or

2. $\mathcal{P}_1 \subseteq \mathcal{N}_2$ and $\mathcal{N}_1 \subseteq \mathcal{P}_2$.

Finally, Murali and Rivera define a *network lego* to be a block $(H, \mathcal{P}, \mathcal{N})$ such that $\sigma_H < \sigma_{H'}$, for every H' where $H \prec H'$ or $H' \prec H$. In other words, $(H, \mathcal{P}, \mathcal{N})$ is a network lego if it is more statistically significant than blocks formed by combining any subset of $\mathcal{P}$ and $\mathcal{N}$ or by combining any superset of $\mathcal{P}$ and $\mathcal{N}$. They output all the blocks that satisfy this condition.

2.4 Conclusions

The algorithms discussed in this chapter integrate *gene expression* data with networks of physical interactions between *proteins*. They make the assumption that the expression level of a gene can be used as a surrogate for the expression or the activity of the protein produced by the gene. This assumption is simplistic, since a single gene may code for multiple proteins due to alternate splicing, and because post-transcriptional and post-translation modifications play a major role in regulating protein levels and activity. Nevertheless, this assumption is very useful in practice, since gene expression does play a major role in controlling physiological process and because DNA microarrays are the most widely-available experimental technology for genome-wide measurement of gene expression.

Chapter 3

Network-Based Enrichment

Christopher L. Poirel, Clifford C. Owens III, and T. M. Murali. Network-based functional enrichment. *BMC Bioinformatics*, 12(Suppl 13):S14, 2011.

3.1 Background

The functioning of a living cell is governed by an intricate network of interactions among different types of molecules. These interactions transduce external signals, control gene expression, protein synthesis and localization, chemically modify protein activities, and drive metabolic and biochemical reactions. Considerable effort in molecular and cellular biology has been expended over the last 50 years by individual research groups on testing and detecting interactions on a small scale. The results of these experiments are enshrined in the literature. In the last few years, a number of efforts have manually curated the literature and created databases of these interactions [55, 59, 103]. More recently, the genomic revolution has inspired the development of experimental technologies that can detect interaction networks in a high-throughput manner and on a genome-wide scale. For example, the yeast 2-hybrid screen has been scaled up to unveil protein-protein interaction networks containing tens of thousands of interactions in a number of organisms [106, 125]. In a similar vein, the chromatin immunoprecipitation on a microarray (ChIP-on-chip) technology allows the detection of the targets of a specified transcription factor on a genome-wide scale [45].

These developments have made molecular interaction networks pervasive in systems biol-

ogy. Concomitantly, a number of computational approaches have been developed to analyze networks and their properties. Foremost among them are methods to reverse engineer gene regulatory networks by integrating gene expression data with other types of -omic data [80]. Such interactions usually relate the expression of a gene to that of other genes in the cell [8]. Another broad class of methods overlay gene expression data for a condition on the wiring diagram to compute the cell's response network for that condition [30, 50, 126].

Networks of these types can contain hundreds or thousands of nodes and an order of magnitude more edges. For example, the B-cell interactome [79, 134], a network of experimentally verified or computationally predicted protein-DNA, protein-protein, and transcription factor-modulator interactions, contains nearly 6000 nodes and over 64,000 edges. It is often desirable to summarize the biological information in such networks. A very common approach is to perform enrichment analysis of the terms in some catalog such as the Gene Ontology [10, 43, 77]. Enriched functions and processes are very useful in summarizing the main biological themes of a large network at a high level, as a preliminary to more detailed mechanistic studies.

When applied to reverse-engineered or response networks, a major drawback of functional enrichment is that it ignores information about the edges in the network being analyzed, i.e., it treats the network simply as a set of genes. Therefore, a function may appear to be enriched in a network, but the genes annotated with that function may be highly disconnected within the network. In such cases, it is difficult to interpret the relevance of that function to the network.

In this chapter, we introduce a novel method for functional enrichment that explicitly takes network interactions into account. Our approach naturally generalizes Fisher's exact test, a widely-used gene set-based technique. We use the sequence of sizes of the connected components of the network to estimate its degree of connectivity. We estimate the statistical significance of this connectivity empirically by a permutation test.

It may be argued that one approach that mitigates the drawback of using gene set enrichment on networks is to find clusters within the network and then compute enriched functions within them. Since clusters are usually densely-connected, enriched functions are likely to induce connected subgraphs within clusters. Our approach is distinct from finding

enriched functions in clusters. Clustering algorithms typically compute dense subgraphs. In contrast, we detect subgraphs (defined by genes annotated with specific functions) that are more connected than may be expected at random. In our results, we show examples of subgraphs that clustering algorithms may not detect. We showcase three applications of our approach:

1. Determine which functions are enriched in a reverse-engineered network.

2. Given a network and a response subnetwork of genes within that network, determine which functions are enriched in the subnetwork.

3. Given two experimentally derived networks, determine the functions for which the connectivity improves when we merge the second network into the first.

These analyses use data from three different species (human, rat, and baker's yeast). They demonstrate that our conceptualization of network-based functional enrichment is a powerful framework for addressing a diverse variety of interesting biological questions about molecular interaction networks.

3.1.1 Formulation of functional enrichment

The problem of functional enrichment is usually formulated as follows. We have a universe U of genes and an "interesting collection" $C \subset U$ of genes. We desire to evaluate the functional coherence of C based on the annotations of the genes in it. To this end, we have access to a set F of biological functions. For each function $f \in F$, let $U_f \subseteq U$ denote the set of genes annotated by f. Furthermore, let $C_f = C \cap U_f$ denote the subset of genes in C that are annotated by f. We will use lowercase letters to denote the cardinalities of the corresponding sets, which are named by uppercase letters. See Figure 3.1 (top) for an illustration.

The standard formulation of functional enrichment in terms of the one-sided version of Fisher's exact test asks the following question:

> If we select a set X of u_f genes uniformly at random (without replacement) from the set of all genes U, what is the probability that $X \cap C$ will contain c_f or more genes?

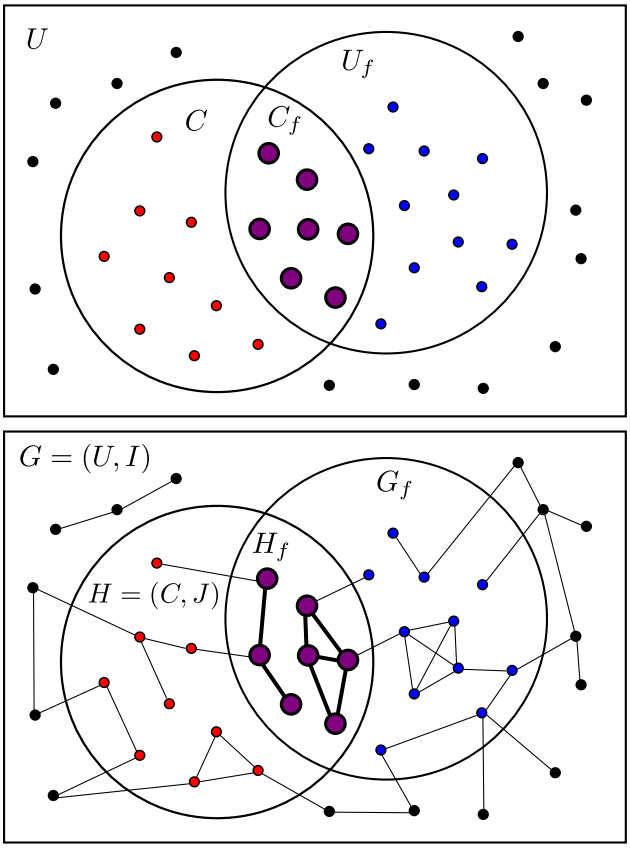


Figure 3.1: (Top) The standard formulation of functional enrichment computes the statistical significance of the size of C_f, the overlap between an interesting collection C of genes and the set U_f of genes annotated with function f. (Bottom) Our network-based approach to functional enrichment computes the statistical significance of the connectivity of H_f, the network induced by the intersection of the interesting collection C and the set U_f of genes annotated by the function.

Thus, we are interested in the probability that a random set of u_f genes would contain c_f or more genes from C. As is well known, we can compute this probability by:

$$h_{\text{Fisher}}(u, c, u_f, c_f) = \sum_{i=c_f}^{\min(c,u_f)} \frac{\binom{c}{i}\binom{u-c}{u_f-i}}{\binom{u}{u_f}} \tag{3.1}$$

Now let us consider the setting in which network-based functional enrichment is relevant. We are given an undirected graph $G = (U, I)$, where U is the set of all genes (as before) and I is a set of edges. We are also given a subgraph $H = (C, J)$ of G, where C (the "interesting collection" of genes) is a subset of U, and J is a subset of I. Note that H may not be the subgraph of G induced by C, i.e., some edges of I that connect node pairs in C may be missing from J. Given a function f, define H_f to be the subgraph of H induced by the genes in $U_f \cap C$, i.e., the subgraph of H induced by the genes that are annotated by f. We desire to use statistics of H_f to estimate whether the function f is enriched in H or not. This concept is illustrated in Figure 3.1 (bottom). Intuitively, even if H_f is highly disconnected, existing methods may declare f to be highly enriched in H. However, it is difficult to interpret the biological relevance of such a function. Therefore, we would like to incorporate the connectedness of H_f into its evaluation.

Ideally, f is highly statistically significant if H_f contains only one connected component, and f is statistically insignificant if H_f contains only singletons (or many small components). We define the *size* of a connected component as the number of nodes in that component. Suppose H_f has m connected components whose sizes are $a_1, a_2, \ldots, a_m$, where $a_i \geq a_{i+1}, 1 \leq i < m$. Using the abbreviation "*cs*" for "component sizes," we will use $cs(H_f)$ to denote this non-increasing sequence of numbers. We would like to estimate the statistical significance of f in terms of $cs(H_f)$. If X is a subset of U, we abuse notation and use H_X to denote the subgraph of H induced by X. Drawing a parallel with the formulation of functional enrichment in the network-free case, we pose the following question:

> If we select a set X of u_f genes uniformly at random (without replacement) from the set of all genes U, what is the probability that $cs(H_X) \geq cs(H_f)$?

Clearly, answering these questions requires that we define how we can compare different values of $cs()$, i.e., decide if one sorted sequence of connected component sizes is "greater

than" another. Note that two distinct subsets X and Y of U may induce different subgraphs of H (i.e., $H_X \neq H_Y$) that have the same sequence of component sizes. Conversely, H_X and H_Y may have the same number of nodes, yet $cs(H_X)$ and $cs(H_Y)$ may be vastly different. (Consider the case where $|X \cap C| = |Y \cap C|$, and H_X is a clique while H_Y is a collection of singletons.) It appears difficult to determine the null distribution of $cs(H_X)$ analytically. Thus, we developed two sampling-based approaches, discussed in "Methods", to compute the statistical significance of $cs(H_f)$ empirically.

3.2 Methods

As before, let $G = (U, I)$ be a network whose nodes are the universal set of genes and let $H = (C, J)$ be a network whose nodes are the interesting collection of genes. We sometimes refer to G and H as the *universal network* and the *interesting network*, respectively. Let H_f be the subgraph of H induced by the nodes annotated with function f.

First, we elaborate on our method for comparing different values of $cs()$, i.e., sorted lists of network component sizes. Next, with a method to compare subnetworks in hand, we proceed to compute the statistical significance of $cs(H_f)$ empirically using two sampling-based approaches.

Comparing sequences of component sizes

Let $A = \{a_1, a_2, \ldots, a_m\}$ and $B = \{b_1, b_2, \ldots, b_n\}$ be sorted sequences such that $a_i \geq a_{i+1}$ for $1 \leq i < m$, and $b_i \geq b_{i+1}$ for $1 \leq i < n$. Since the a_i's and b_i's represent component sizes, we assume all values are positive. If $m < n$, we pad A with zeros by setting $a_i = 0$ for $m < i \leq n$. We pad B similarly in the case that $n < m$.

Now, we naturally define $A = B$ if and only if $a_i = b_i$ for all $1 \leq i \leq \max(m, n)$. Otherwise, we define $A < B$ if and only if there exists some index i, $1 \leq i \leq \max(m, n)$, such that $a_i < b_i$, and $a_j = b_j$ for all $j < i$. If neither of these cases hold, we say that $B < A$. Essentially, we walk along A and B simultaneously (which have the same length after padding) until we find an index i where $a_i \neq b_i$. The smaller sequence is the one that contains the smaller of those two values.

3.2.1 Function randomization

The function randomization approach for computing the statistical significance of $cs(H_f)$ parallels the sampling-based alternative to the analytical solution for the one-sided version of Fisher's exact test (Equation 3.1). In the sampling-based solution for Fisher's exact test (that is not network-based), we repeatedly select a set X of u_f genes uniformly at random from the universe of genes U. We then calculate an empirical p-value for the function f, which represents the statistical significance of the size of C_f, as the fraction of samples for which $|X \cap C| \geq c_f$, i.e., the size of the intersection between the randomly selected set X and the interesting collection of genes is at least as large as C_f. If we repeat this process many times, the empirical value converges to the analytical value computed by Equation 3.1.

In the case of network-based enrichment, we have not been able to derive the null distribution of $cs(H_X)$ analytically. Therefore, we apply a sampling-based algorithm similar to the one used in the non-network case. We repeatedly select a set X of u_f genes uniformly at random from the universal genes U. At each iteration we compute the subgraph of $H = (C, J)$ induced by genes in $X \cap C$ and call this subgraph H_X. We calculate the p-value for f as the fraction of random choices of X for which $cs(H_X) \geq cs(H_f)$. This p-value is an empirical estimate of the probability that the intersection between the interesting network and a randomly-selected subgraph of the universal network is more well-connected than H_f.

This network-based formulation of functional enrichment strongly parallels the traditional non-network formulation. In fact, notice that if we remove all edges from G and from H, the network-based solution is exactly the same as the one-sided version of Fisher's exact test. In this sense, we generalize a standard functional enrichment approach for gene sets to networks induced by sets of genes.

3.2.2 Network structure randomization

The function randomization approach for computing network-based functional enrichment tests the dependency of $cs(H_f)$ on the size of the set of genes annotated with f. However, H_f also depends on the specific interactions present in the underlying universal network $G = (U, I)$. To test the dependence of $cs(H_f)$ on G, we developed a separate method that relies on

randomizing the structure of G. We use the following algorithm [82] to generate a randomized universal network with the same degree sequence as G, i.e., the degree of each node is the same in G and in the new network. This algorithm preserves topological properties of molecular interaction networks, such as their scale-free degree distribution.

Algorithm 3.1 Edge-swap randomization algorithm

EdgeSwap$(G = (U, I), k)$

$I' \leftarrow I$
for $1 \leq i \leq k|I|$ **do**
  Randomly select two edges $(a, b), (c, d) \in I'$
  if $a = d$ or $b = c$ **then**
    Do nothing
  else if $(a, d) \in I'$ or $(b, c) \in I'$ **then**
    Do nothing
  else
    $I' \leftarrow I' \setminus \{(a, b), (c, d)\}$
    $I' \leftarrow I' \cup \{(a, d), (b, c)\}$
return $G' = (U, I')$

The first *if* statement in EdgeSwap prevents the algorithm from producing self-loops in the random network. The second *if* statement ensures that the edges to be added do not already exist in I', thereby keeping the resulting graph simple. We perform the edge swap randomization for $k|I|$ iterations, where k is an external parameter to this algorithm. We set $k = 10$ for all analysis presented in this paper. We determined this value for k by running EdgeSwap with multiple values of k on the networks discussed in Section 3.3. For each value of k, we analyzed the distribution of the overlap between the edges in a random network and the original network. Increasing k beyond 10 did not show any significant change in this distribution (compared to $k = 10$) for any of the original networks.

In order to assess the significance of $cs(H_f)$ for each function, we repeatedly generate a randomized universal network $G' = \text{EdgeSwap}(G, k)$. Let $H' = (C, J')$ be the subgraph induced in G' by the genes in C, and let H'_f be the subgraph of H' induced by the genes annotated with function f; H' and H'_f are subgraphs of G' in the same fashion as H and H_f are subgraphs of G. We calculate the p-value for each function f as the fraction of iterations (i.e., random networks) for which $cs(H'_f) \geq cs(H_f)$. This p-value indicates the probability that the genes in H_f are more connected after randomly shuffling the edges in G (including

those in H_f) while maintaining the degree sequence of G.

3.2.3 Combining p-values

We developed the function randomization approach because of its strong ties to the standard way we interpret functional enrichment. Furthermore, it can be expressed as an extension of the one-sided version of Fisher's exact test. This approach is an attempt to answer the following question: "Given a function f that annotates u_f genes, what is the probability that a randomly-selected set of u_f genes will have a more connected intersection with the interesting genes than H_f does?"

As an alternative, we developed the network structure randomization approach because we were interested in a strictly network-based formulation of functional enrichment. This approach is an attempt to answer a very different question from the previous method: "Given a function f, what is the probability that the subgraph induced by genes in C_f will be more connected in a network chosen uniformly at random from the set of all networks with the same degree sequence as G?"

In our analyses, we only consider those functions deemed well-connected by *both* approaches. Thus, we assign a p-value for each function that is the maximum of the two p-values from each of the two approaches. In the rest of the paper, unless stated otherwise, the p-value for a function refers to the maximum of the p-values computed by both approaches.

3.3 Results

We present applications of our methods to three different interaction networks, each for a different organism. In the first application, we identify biological processes from the Gene Ontology (GO) [6] with significant network-based enrichment in the human B cell interactome [72]. In the second application, we discuss gene sets from the Molecular Signatures Database [118] that are deemed significantly enriched in a response network that represents the differences between collagen sandwiches and hepatocyte monolayers, two widely-used systems to culture rat hepatocytes [66]. In the third application, we discover GO biological

processes whose network-based enrichment improves when we add a collection of genetic interactions in *S. cerevisiae* that were identified in a large-scale study conducted by Krogan *et al.* [25] to the BioGRID network [116].

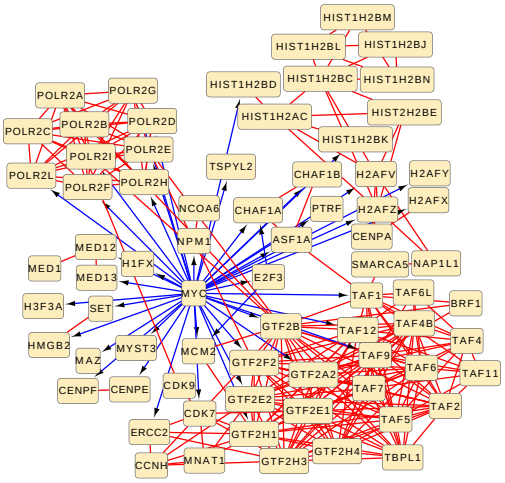
Each of these three applications of network-based enrichment showcases how our methods can be used in a different manner: *i*) determine which functions are enriched in a given network, *ii*) given a network and an "interesting" subnetwork of genes within that network, determine which functions are enriched in the subnetwork, and *iii*) given two networks, determine the functions whose connectivity improves by merging the second network into the first. These three applications ask different questions about the functional enrichment of an interaction network, each of which we can answer using network-based enrichment.

We used the Synergizer [14] to generate all gene and protein identifier mappings. We visualized all networks using Cytoscape [111].

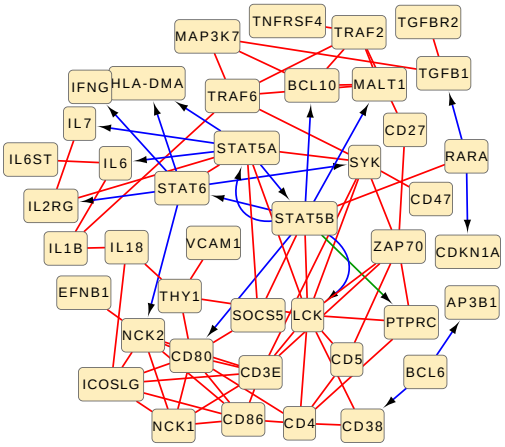
3.3.1 B cell interactome

To analyze functions enriched in human B cells, we applied our methods to the B cell interactome (BCI) [72], which incorporates three different types of interactions: protein-protein, protein-DNA, and transcription factor-modulator. The BCI includes both experimentally verified and computationally predicted interactions [79, 134] in human B cells. Protein-DNA and transcription factor-modular interactions are directed. For the purposes of this work, we treated them as being undirected. We removed repeated edges and self-loops from the BCI, resulting in an interaction network with 5748 nodes and 64,007 edges. We applied our network-based enrichment methods to identify GO biological processes enriched in the BCI. After downloading functional annotations from the GO website [6], we applied the true-path rule to the annotations, i.e., if a gene was annotated with function f, we ensured that it was also annotated with any ancestors of f in the GO directed acyclic graph.

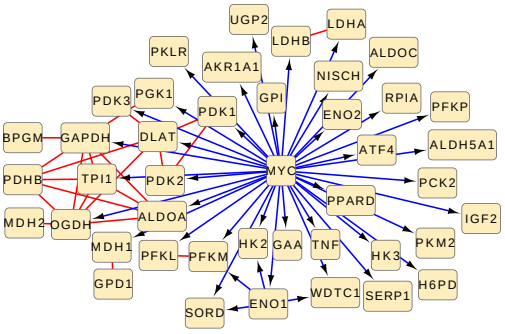
Recall that our enrichment methods require a universal network G and an interesting network H. Since we were simply interested in the set of functions enriched in the BCI, we let both the universal network and the interesting network be the BCI. Notice that the standard formulation of gene set enrichment (Equation 3.1) will assign a p-value of 1 to every function, since the size of the interesting set of genes and the size of the universe are



(a) Protein-DNA Complex Assembly



(b) Positive Regulation of Lymphocyte Activation



(c) Glucose Metabolic Process

Figure 3.2: Subnetworks of the BCI induced by genes annotated with (a) GO:0065004 Protein-DNA Complex Assembly, (b) GO:0051251 Positive Regulation of Lymphocyte Activation, and (c) GO:0006006 Glucose Metabolic Process. The red, blue, and green edges represent protein-protein, protein-DNA, and transcription factor-modulator interactions, respectively.

equal (i.e., $u = c$ and $u_f = c_f$). As a result, we cannot compare our method for discovering enriched functions to the set-based Fisher's exact test for this application. However, our approach assigns non-trivial p-values to several relevant functions, as we show below.

We computed p-values by executing each random sampling approach 100,000 times. We retained those functions that annotated at least five genes in the universal network and no more than 100 genes in the interesting network, resulting in a collection of 2098 GO biological processes. Of these, 31 processes received a network-based enrichment p-value of less than 10^{-5}, which is the smallest empirical p-value possible over 100,000 iterations. Next, we present three significantly-enriched functions and discuss their relationship to the B cell interactome. The goal of this analysis is to demonstrate that network-based enrichment methods were capable of finding meaningful functions in a network associated with a specific cellular context. Figure 3.2 illustrates the subnetwork induced by the genes annotated with each of the three functions. In other words, each of the networks in Figure 3.2 is a visualization of the subgraph H_f corresponding to one of three discussed functions. These figures only retain nodes that are incident on at least one edge. While we do not incorporate the directionality of the protein-DNA and transcription factor-modulator interactions in our methods, we considered the directionality when visualizing the results.

We discuss functions that were ranked 1, 32, and 43 overall. Note that 31 processes received the smallest p-values. Hence, all of them have the rank of 1. We observed that a majority of these functions were related to protein phosphorylation and kinase signaling cascades. The term we discuss next is one that we could relate to B-cells based on the proteins annotated by that term.

Protein-DNA complex assembly

The GO term "Protein-DNA Complex Assembly" (GO:0065004) was enriched in the BCI with a p-value of less than 10^{-5} (rank 1 out of 2098 functions). Figure 3.2 demonstrates that the genes annotated by this function form several densely-connected subgraphs. A clustering algorithm may have detected each individual subgraph but may not have included all of them in a single cluster. Thus, the term "Protein-DNA Complex Assembly" may be determined to be enriched (using set-based approaches) in the results of a clustering algorithm only if

the clusters were themselves further grouped during a post-processing procedure.

The genes in the subgraph induced by this function are associated with various aspects of the aggregation and binding of proteins with DNA molecules to form a protein-DNA complex. For example, the nine HIST and four H2A proteins are grouped together. DNA wraps around the histones, forming higher-order complex protein-DNA subunits. Many eukaryotic genes contain a DNA signature in their promoter region known as the TATA box. These genes rely on the ordered assembly of RNA polymerase II and several initiation factors at the TATA box. The ten POLR genes that form a clique encode different subunits of RNA polymerase II. The eight densely-connected general transcription factors (GTF2 genes in Figure 3.2) serve as initiation factors for RNA polymerase II. The eleven neighboring TAF genes perform a similar role. This function was likely determined to be enriched because of the hub transcription factor MYC, which serves as a bridge between these densely-connected subnetworks. Since MYC is responsible for regulating many other genes, irregular expression of MYC has been linked to several cancers, and some studies have identified this gene as a potential cancer drug target in humans [115].

Lymphocyte activation

We found the term "Positive Regulation of Lymphocyte Activation" (GO:0051251) enriched in the BCI with a p-value of 10^{-5} (rank 33). We expected to discover this function as enriched in the BCI: since B cells are a specific type of lymphocyte, up-regulation of lymphocyte activation is an inherent property of genes in the BCI. Our methods were capable of determining not only that many genes related to lymphocyte regulation appear in the BCI, but that there was a rich interconnectivity among these genes. Figure 3.2 illustrates that the signal transducer and activator of transcription (STAT) proteins (STAT5A, STAT5B, STAT6) are central to this connectivity. STAT5 has been shown to play a key role in the development and proliferation of B cells [47, 78] through its activation by and mediation of many interleukins (e.g., IL2, IL6, and IL7, which are also present in this network). Each of these interleukins serves as a growth factor for various B cell lineages.

Glucose metabolism

We found the term "Glucose Metabolic Process" (GO:0006006) enriched in the BCI with a p-value of 2×10^{-5} (rank 43). While glucose metabolism is not solely related to the behavior of B cells, this process is important for many cell types, including B cells, as a primary source of energy. Perhaps what was most interesting about this function was its reliance on MYC for most of its connectivity. As mentioned, MYC is considered to be a master regulator because it regulates the activity of a large number of human genes. Figure 3.2 demonstrates that without the annotation of MYC by "Glucose Metabolic Process" the network would consist of mostly disconnected proteins. While MYC also plays a central role in "Protein-DNA Complex Assembly", one could argue that, even without MYC and the edges incident on it, the protein-DNA complex assembly network is reasonably well-connected and will be deemed to be significantly enriched by our methods. In the case of "Glucose Metabolic Process", we elucidated a function whose network-based enrichment score relies heavily on its connectivity through a central hub.

3.3.2 Hepatic cultures

The liver carries out a multitude of necessary functions in humans and many other animals, including the metabolism of foreign compounds (xenobiotics) and cholesterol. Hepatocytes constitute roughly 70-80% of the liver cells. Two commonly used systems for culturing these cells *in vitro* are the hepatocyte monolayer (HM) and the collagen sandwich (CS). Briefly, the HM consists of a layer of collagen on top of which a single layer of hepatocytes are placed; the CS is similar to the HM with the addition of an extra layer of collagen on top (creating a "sandwich" of hepatocytes between two layers of collagen). A recent study analyzed the expression profile over eight days of genes in primary rat hepatocytes in both HM and CS tissue cultures [66]. The authors demonstrated that over eight days, the genes annotated by a wide range of liver-specific biological processes were consistently up-regulated in hepatocytes in CS but not in HM and that processes related to the cell cycle were down-regulated in CS, as compared to HM. Hepatocytes in HM differentiate and lose some of their morphology over time, contributing to their loss of liver-specific functions.

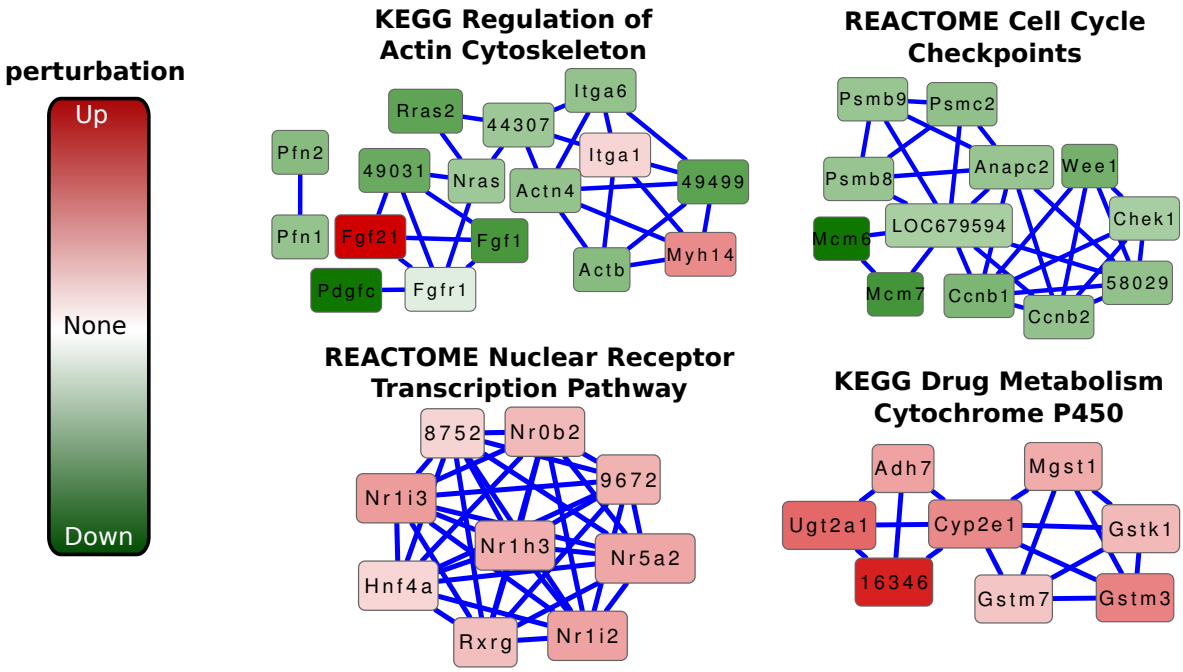


Figure 3.3: Subnetworks of a hepatocyte response network induced by genes annotated with MSigDB gene sets. (Top Left) (KEGG) Regulation of Actin Cytoskeleton, (Top Right) (REACTOME) Cell Cycle Checkpoints, (Bottom Left) (REACTOME) Nuclear Receptor Transcription Pathway, and (Bottom Right) (KEGG) Drug Metabolism Cytochrome P450. Red and green nodes indicate up- and down-regulation, respectively, of individual genes in collagen sandwich versus hepatocyte monolayer tissue cultures. Darker node color indicates higher perturbation in either direction, as indicated by the legend on the left.

For this application of our enrichment methods, we sought to discover functions that annotate differentially-expressed genes in the CS cultures when compared to HMs. We used the STRING database of known and predicted protein interactions [121] as our universal interaction network. STRING assigns a weight between 0 and 1000 to each interaction based on the combined scores of various sources of evidence for the interaction, where higher scores indicate more confidence for that interaction. We removed any self-loops or interactions with a score below 500 from the universal network, resulting in a network of 9925 nodes and 204,992 edges. We analyzed the expression profiles for rat hepatocyte genes in HM versus CS cultures after 8 days of growth [66], since CS showed the greatest divergence from HM after 8 days. Based on these expression profiles, we used LIMMA [114] to compute p-values that represented the significance of the differential expression of each gene between CS and HM. We desired to compute the response network (a subgraph of the STRING network) that captured the differential expression between CS and HM cultures. Accordingly, we used these per-gene p-values in concert with the BioNet algorithm [12] to compute a response network (within the STRING universal network) of interconnected genes that is perturbed in CS, in comparison to HM. We used the default parameters for BioNet, with FDR=0.001. Please see the BioNet publication for more details. The response network contained 876 nodes and 3423 edges. We collected functional annotations of the genes in this network from the Molecular Signatures Database (MSigDB) [118]. We only retained curated gene sets (category C2) and GO gene sets (category C5) from MSigDB, and we removed any functions that annotated fewer than five genes in the universal interaction network.

With the universal interaction network, an interesting subnetwork of the universe (i.e., the BioNet response network), and functional annotations of the universal genes, we applied our network-based enrichment methods. We tested the enrichment of 4035 MSigDB functions for 100,000 iterations. We also computed the enrichment of each function, using an empirical version of Equation 3.1, to allow us to compare our network-based enrichment p-value for a given function to the p-value given by a standard gene set-based enrichment method (i.e., the one-sided version of Fisher's exact test). We did not use the analytical version of Fisher's test so that the enrichment p-value estimated by our methods for each function was comparable to that computed by Fisher's test. We observed that 177 of the 4035 functions received

a network-based enrichment p-value of less than 10^{-5}, the smallest possible empirical p-value for 100,000 iterations (i.e., our methods returned a p-value of 0 for these functions). We present four of the top 177 functions and analyze their relationship to liver cultures. Figure 3.3 illustrates the networks induced by the genes annotated by these four top-ranking network-based enriched functions (i.e., we show H_f for each function). The genes are colored based on their level of perturbation in the contrast between CS and HM. Lightly-colored genes indicate little or no perturbation, while dark red (green) nodes denote significant up- (down-) regulation in CS compared to HM.

Actin cytoskeleton

Our network-based enrichment methods discovered the KEGG [59] pathway "Regulation of Actin Cytoskeleton" enriched with a p-value less than 10^{-5} (rank 1). Comparatively, the empirical p-value for the one-sided version of Fisher's exact test was 0.02857 (rank 819), more than three orders of magnitude larger than our p-value. Actin organizes into thin filaments to provide structure to cells. Cell locomotion is often a driving force for the development of actin cytoskeleton. Actin filaments form a highly-organized, complex structure, and manipulation of the actin cytoskeleton enables adhesion to the substrate and movement of the cell [135]. However, hepatocytes are generally stationary cells and are unlikely to use this mechanism. More specifically, HM hepatocytes lose the true hepatic phenotypes much faster than those in CS cultures. A specific characteristic of this loss is the production of actin fibers that help cells in the HM culture to better adhere to the underlying substrate. This phenomenon explains why many genes in the "Regulation of Actin Cytoskeleton" network (Figure 3.3) are consistently down-regulated in CS hepatocytes compared to those in HM. Note that this network contains some up-regulated genes, including Fgf21 and Myh14. These genes are involved in a wide variety of biological processes including cell growth, tissue repair, and cell polarity. Their participation in several processes not directly related to actin development may explain why they are upregulated in CS compared to HM in this network.

Cell cycle checkpoints

We found the REACTOME [55] pathway "Cell Cycle Checkpoints" enriched with a p-value less than 10^{-5} (rank 1), while the empirical version of Fisher's method assigned a p-value of 0.05539 (rank 1034). Many genes in this network are essential to the cell cycle process. ANAPC2 codes for part of the anaphase-promoting complex, which is responsible for promoting eukaryotic cells from metaphase to anaphase during mitosis. Wee1 serves as a key checkpoint in the cell cycle by inhibiting entry into mitosis until the cell has grown to a certain size, thus preventing the separation of the cell into two daughter cells that are too small. Mcm6 and Mcm7 are essential genes for DNA replication in eukaryotes, a significant process in the cell cycle. Under healthy conditions, mature hepatocytes do not readily divide except to replace damaged hepatocytes [84]. This observation may explain the downregulation of these genes in CS compared to HM. In other words, down-regulation of cell cycle checkpoints prevents the cell cycle from progressing in CS cultures, in comparison to HM cultures.

Nuclear receptor transcription pathway

We identified the REACTOME "Nuclear Receptor Transcription Pathway" enriched with a network-based p-value less than 10^{-5} (rank 1). Fisher's method returned a p-value of 0.0092 (rank 565). Nuclear receptors are responsible for sensing steroids, hormones, and other signaling molecules in the cell. Since one of the primary functions of the liver is the production of hormones, enrichment of this gene set in CS cultures is to be expected and was noted in an earlier study [66]. Furthermore, we expect many genes involved in this pathway to exhibit up-regulation in CS versus HM cultures. Figure 3.3 confirms that all the genes in the intersection between this pathway and the BioNet response network are indeed up-regulated. Nr1h3, Nr1i2, Rxrg, and Hnf4a encode parts of the liver X receptor, the pregnane X receptor, the retinoic acid receptor, and the hepatocyte nuclear factor, respectively. All these nuclear receptors are responsible for sensing signaling molecules that are highly relevant to liver function, e.g., metabolism of toxic substances and vitamins.

Drug metabolism

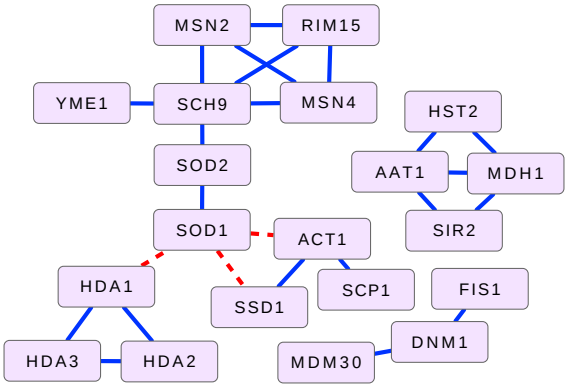
The KEGG pathway "Drug Metabolism of Cytochrome P450" was enriched according to our network-based method with a p-value of less than 10^{-5} (rank 1). Fisher's method assigned the same function a p-value of 0.01572 (rank 681). All genes in this network were up-regulated in CS hepatocytes compared to those in HM. One of the primary functions of the liver is processing xenobiotics, including drugs. Cytochrome P450s represent a class of proteins that are responsible for breaking down various lipids, steroids, and xenobiotic (external) compounds. Primary hepatocytes actively express these proteins, even in the absence of external chemicals or drugs [48]. The enrichment of this pathway in CS cultures (in comparison to HMs) supports the well-known phenomenon that HMs rapidly lose liver-specific functions.

Gene sets with high network-based p-values

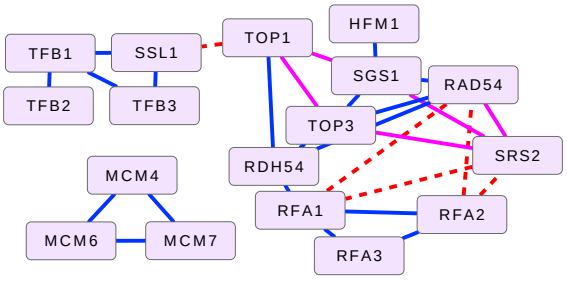
Several functions received a high network-based p-value from our methods and a low set-based p-value using Fisher's exact test. We ranked the genes sets in increasing order of their set-based p-values, and we report the two highest-ranking gene sets that received a network-based p-value of 1. According to Fisher's exact test, we found the MSigDB gene set "Su Liver" significantly enriched with a p-value of 1.77^{-6} (rank 89) and the gene set "Chiaradonna Neoplastic Transformation Kras DN" significantly enriched with a p-value of 2.48×10^{-6} (rank 93). Our network-based functional enrichment approach deemed both functions insignificant with a p-value of 1 because there exist no interactions among the genes annotated by either function. Gene sets whose constituent proteins are sparsely-connected may be difficult to interpret.

3.3.3 Improving functional network coherence

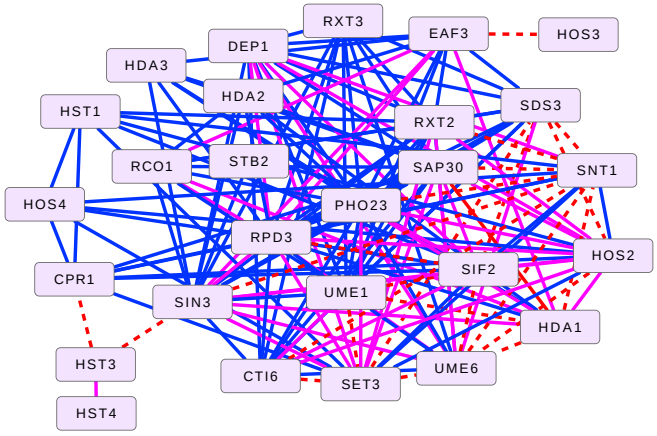
In the third and final application of our network-based enrichment methods, we sought to analyze how the addition of a collection of experimentally determined interactions impacts a dataset of known interactions. Specifically, given a universal interaction network in *S. cerevisiae*, we wanted to determine which biological functions became more coherent within



(a) Chronological Cell Aging



(b) DNA Geometric Change



(c) Histone Deacetylation

Figure 3.4: Subnetworks of the BioGRID universal network induced by genes annotated with GO biological processes (a) GO:0001300 Chronological Cell Aging, (b) GO:0032392 DNA Geometric Change, and (c) GO:0016575 Histone Deacetylation. Blue interactions are edges in BioGRID that were not identified by the GI study [25]. Solid pink edges were identified in the GI study and were also present in BioGRID through some alternative evidence. Dashed red interactions were only discovered by the GI study.

the network after introducing new pairwise interactions discovered in a large-scale genetic interaction (GI) study. We retrieved the yeast interactome from the BioGRID database [116], which incorporates both genetic and physical interactions from multiple sources (5681 nodes and 97,862 edges). BioGRID includes a subnetwork of 14,421 GIs among 721 genes reported by Collins *et al.* [25]. We refer to this list of GIs as "the GI study" or "the GIs" throughout this section. Our goal was to ask which GO biological processes became better connected when the GIs were added to BioGRID. We downloaded GO biological process annotations of the genes in BioGRID from GO [6]. As with the annotations of the BCI, we applied the true-path rule to the annotations, and we removed any functions that annotated fewer than five genes in the universal network and more than 100 genes in the interesting collection C.

We then applied our network-based enrichment approach in two ways. The universal interaction network (G) remained the same in both cases. In the first case, we set the interesting network (H) to be the same as the universal network (which also includes edges from the GI study). This analysis is similar to the analysis performed on the BCI. Essentially, the network-based p-values in this case indicated which functions were enriched in the entire BioGRID network, including the GIs. For the second case, we set the interesting network to be the universal network with interactions from the GI study removed. If there were other sources of evidence for one of the GIs, we retained that edge in the interesting network. The network-based p-values in this case indicated which functions were enriched in the universal network if we ignored any information from the GI study.

Let $p_a(f)$ be the network-based p-value for function f from the first case, where we considered *all* edges in the universal network, including those from the GI study. Similarly, let $p_r(f)$ be the network-based p-value for function f from the second case, where we *removed* any edges in the GI study from the interesting network. We scored each function by $s(f) = \log_{10}(p_r(f)/p_a(f))$. Thus, functions whose network-based enrichment p-value decreased (became more significant) as a result of adding the GIs received a positive score, functions whose enrichment remained the same received a score of 0, and functions whose p-values increased (became less significant) as a result of adding the GIs received a negative score. We believed that highly positive-scoring functions were those biological processes whose coherence improved the most by adding the GIs. Of the 1443 functions tested, 28

functions received a score greater than 1 (i.e., the p-value decreased by more than 1 order of magnitude after adding the GIs), and only 1 function received a score less than -1 (i.e., the p-value increased by more than 1 order of magnitude by including the GIs). Next, we discuss the three GO biological processes, "Chronological Cell Aging", "DNA Geometric Change", and "Histone Deacetylation", that received the 2nd, 3rd, and 9th highest overall scores, respectively. These examples illustrate a new application of network-based functional enrichment that is sensitive to the increased connectivity among a set of genes upon the addition of multiple edges. However, gene set-based enrichment may not be sensitive to such an addition of edges if, for example, all newly-added edges were among genes already present in the network.

Cell aging

The GO biological process "Chronological Cell Aging" (GO:0001300) received a score of 2.6464 ($p_a = 0.0003$ and $p_r = 0.1329$) exhibiting a decrease in network-based p-value of nearly three orders of magnitude with the addition of the GIs. This process is the progression of quiescent (non-dividing) cells from their inception to the end of their lifespans. Figure 3.4 demonstrates that this increase in enrichment is a result of three GIs incident on SOD1, a Cu-Zn superoxide dismutase that has a role in the detoxification of oxygen radicals. These enzymes catalyze the breakdown of the superoxide radical (O2-) into an oxygen molecule and hydrogen peroxide. Yeast cultures lacking SOD1 exhibit drastically decreased viability [76]. Strikingly, the interactions from the GI study were able to connect three previously disconnected subnetworks of genes annotated with chronological cell aging in *S. Cerevisiae*.

DNA geometric change

Our approach identified the GO term "DNA Geometric Change" (GO:0032392) with a score of 2.32 as having a considerably decreased network-based enrichment p-value after adding the GIs (p_a less than 10^{-4}, but $p_r = 0.0021$). The p-value decreased primarily because of the GI between SSL1 and TOP1, which connected two previously disconnected components of the network induced by genes associated with DNA geometric change. Additionally,

the added GIs reinforced the connectivity among RFA1, RFA2, RAD54, SRS2, and their neighboring genes. Thus, the connected component induced by these genes was less likely to become disconnected during the randomization procedures used to perform network-based enrichment.

Histone deacetylation

Our enrichment approach assigned a score of 1.5955 ($p_a = 0.0005$ and $p_r = 0.0197$) to the GO term "Histone Deacetylation" (GO:0016575). The GIs improved the connectivity among several genes related to histone deacetylation (HDA1, HOS2, SDS3, SIF2, SNT1, UME1, and UME6). The GIs also provided a new source of evidence for many previously-known interactions, a property we can deduce from Figure 3.4 but one that is not explicitly considered by our methods. However, the interaction connecting HOS3 to EAF3 and the interactions between HST3 and CPR1/SIN3 are perhaps the most interesting as they provide new evidence for including HOS3, HST3, and HST4 in the large connected component associated with histone deacetylation. The GIs that connect HST3 and HST4 to the largest connected component of this biological process are quite striking, since these genes code for histone deacetylases, and the GIs provide previously unknown interaction data for including them in this network.

3.4 Conclusions

In this chapter, we presented a novel approach to functional enrichment that takes into account the pairwise relationships among genes annotated by a particular function. We proposed two different methods for calculating p-values to assess the significance of each function in a network-based context; we described these methods as "function randomization" and "network structure randomization". We required functions to have low enrichment p-values based on both criteria. Our function randomization approach is a generalization of the one-sided version of Fisher's exact test, a standard formulation of functional enrichment for gene sets. Specifically, after removing edges from the universal and interesting networks, the function randomization approach is equivalent to the one-sided version of Fisher's exact

test. Our network structure randomization approach offers a strictly network-centric method for determining functional enrichment.

We utilized our methods on real biological data from three different organisms: human, rat, and yeast. In each organism, we showcased a different application of network-based enrichment. First, we used the human B cell interactome to demonstrate the capability of our methods to simply discover enriched functions in a single network. We have noted that many standard gene set enrichment methods do not address this issue, as they require a universal gene set and an interesting collection of genes within the universal genes. Moreover, graph clustering algorithms that are designed to compute dense subgraphs might not detect subnetworks that are well-connected but not very dense, such as those corresponding to "Protein-DNA Complex Assembly" and "Glucose Metabolic Process" in Figure 3.2. Second, we applied our methods to a response network composed of rat genes perturbed in hepatocytes cultured in two different ways. This approach parallels traditional gene set enrichment methods, where given an interesting collection of genes, we seek functions overrepresented in the interesting genes with respect to some universe of genes. We identify several relevant functions that our network-based method finds as being most significant but are far from the highest-ranking functions using the set-based Fisher's method. Third, we demonstrated a novel application of network-based enrichment to assess the functional contribution of a collection of genetic interactions in yeast to the underlying universe of known yeast interactions.

Our work suggests many directions of future research. While we did not take edge reliabilities into account for the analysis presented here, our methods can be readily applied to weighted networks. Indeed, our approach for comparing sequences of component sizes can be used to compare sorted sequences of the sums of edge weights within each component (rather than the number of nodes). Identifying alternative methods for sequence comparison is also an interesting question, particularly those methods for which the resulting distribution of sorted sequences can be determined analytically. Such comparison methods may drastically improve the computational efficiency of our network-based enrichment approach. Our methods analyze the functions one at a time and often lead to redundant functions being identified as enriched. Additionally, we need to apply corrections for multiple hypotheses testing,

thereby potentially decreasing our statistical power. Furthermore, our permutation-based approach for computing p-values is very time-consuming. These three issues can potentially be tackled by generalizing model-based approaches for gene function enrichment [10, 77] to the domain of network-based function enrichment. After assuming an appropriate model for how biological processes may be perturbed in a cell, these methods proceed to infer the set of biological processes that best explain an input list of genes (the interesting collection) against a background list (the universe). Applying these methods in the network context is an interesting and important problem.

Chapter 4

Reconciling differential gene expression data with molecular interaction networks

Christopher L. Poirel, Ahsanur Rahman, Richard R. Rodrigues, Arjun Krishnan, Jacqueline R. Addesa, and T. M. Murali. Reconciling differential gene expression data with molecular interaction networks. *Bioinformatics*, 29(5):622–629, March 2013.

4.1 Background

A cell's response to its environment is governed by an intricate network of molecular interactions. These interactions dynamically change in response to a myriad of cues. Variation in the expression profile of the cell contributes greatly to the collection of context-specific molecular interactions. For example, the transcriptional program in brain cells from a patient diagnosed with Alzheimer's disease may vary greatly from that of a patient diagnosed with Huntington's disease. Differences in gene expression influence the abundance and activities of their gene products and the physical and functional interactions among them. Therefore, discovering the response network, i.e., the set of molecular interactions that are active in a given cellular context is a fundamental question in computational systems biology [51].

Many response network algorithms integrate molecular interaction networks with treatment-

control differential expression data, which quantifies the statistical significance of the difference between the expression of genes under two conditions, e.g., diseased versus healthy cells [12, 61, 101, 127]. Given several treatment and control samples, these methods compute a p-value for each gene that indicates the statistical significance of its differential expression between treatment and control. These approaches typically integrate such expression data with the interaction network by directly using each gene's p-value or some transformation of the p-value as the weight of the gene in the network.

Ideker *et al.* pioneered this type of analysis. After converting per-gene p-values into z-scores, and weighing each node in the network accordingly, they defined the score of a given subgraph as a Liptak-Stouffer z-score, i.e., the sum of the z-scores of the nodes in the subgraph divided by the square root of the number of nodes in the subgraph. They subsequently computed connected subgraphs with high aggregate z-scores. Beisser *et al.* extended the approach of Ideker *et al.*, by solving the prize-collecting Steiner problem. Chowdhury *et al.* identified dysregulated subnetworks by computing state functions that indicate which combination of up- and down-regulated genes within the subnetwork can classify the disease of interest. The goal of many of these approaches is to score genes by combining their expression profiles across multiple samples and subsequently compute a (connected) subgraph such that the genes in it jointly optimize some combination of their scores. A recent trend, exemplified by BioNet [12] and DEGAS [127], is to focus on computing subgraphs with few nodes. The rationale behind these approaches is that small subnetworks enriched with dysregulated genes correspond to parts of the disease-related pathways. Such approaches were motivated by the observation that only part of a pathway is often changed by disease [127] and that interpreting small subnetworks is easier than interpreting larger ones [12]. HotNet [129] identifies significantly mutated pathways in cancer by marking mutated genes in an interaction network and propagating this information throughout the network by a method similar to the Heat Kernel (see Methods).

While these recent methods are powerful at highlighting specific pathways and subnetworks, they have not been designed to uncover the spectrum of processes and pathways that might be perturbed in a particular experiment or disease. Applying functional enrichment tests to significantly differentially expressed genes can address this issue, but such analysis

typically ignores underlying protein interactions. Top-ranking differentially expressed genes are often highly disconnected in the corresponding protein interaction network, making it difficult to discern the precise mechanisms by which enriched pathways affect the disease. Furthermore, insignificantly differentially expressed genes may represent crucial components of disease-related pathways, but such genes are often ignored by standard enrichment methods.

Motivated by these observations, rather than computing condition-specific subgraphs, we propose to use previously-published approaches that *reconcile* differential gene expression p-values with an underlying interaction network in order to re-rank *all* genes with respect to the treatment of interest. We describe techniques developed in the machine learning and information retrieval communities [23, 95, 146, 147] to reconcile gene expression data with the interactions in the network by computing smooth functions over neighboring nodes in the underlying network. In this paper, we use the negative logarithm of the per-gene p-values (which do not directly incorporate any interaction data) as prior indicators of each gene's relevance to a particular treatment; we describe mathematical functions that allow the expression values to change so that two genes connected by an interaction have similar values, while controlling the deviation of those values from their original settings. This modification allows genes with no significant differential expression to be re-ranked highly if their products interact with the products of many significantly differentially expressed genes.

This work offers the following three novel contributions. First, we propose that such reconciliation algorithms should strive to maintain the following properties:

(i) Top-ranking genes after reconciliation should participate in coherent network structures, i.e., interacting genes should receive similar scores. This property can assist in the functional evaluation of top-ranking genes.

(ii) Reconciled gene rankings given by different treatments should be dissimilar, especially among top-ranking genes. This requirement ensures that the process of reconciliation does not dilute the differences between the transcriptional signatures of distinct diseases or treatments.

(iii) Top-ranking reconciled genes should be functionally coherent. For biological pathways

that are relevant to a disease, low-ranking genes annotated by those pathways should be re-ranked highly by the reconciliation process.

Second, we comprehensively assess the extent to which each algorithm addresses these three properties. Third, we evaluate each approach on a large compendium of gene expression data that includes 54 diverse human diseases. We use this assessment to identify an ideal value for the input parameter required by each of the four algorithms. We apply a state-of-the-art functional enrichment algorithm [10] to address the functional coherence of the gene rankings provided by the reconciliation algorithms. We demonstrate that the reconciled gene rankings identify disease related functions that would be missed by analyzing statistically significant differentially expressed genes alone.

4.2 Comparison to previous approaches

Here we provide a qualitative comparison of network reconciliation methods to techniques that have already been published in the literature. As mentioned previously, BioNet [12] and DEGAS [127] aim to compute small connected subgraphs that are correlated with a disease. Such approaches were motivated by the observation that only part of a pathway is often changed by disease [127] and that interpreting small subnetworks is easier than interpreting larger ones [12]. We differ from these techniques in the following ways:

(i) DEGAS and BioNet formulate problems that are computationally intractable (NP-complete or NP-hard). They solve these problems either by using approximation algorithms or methods to solve integer linear programs. In contrast, we propose comparatively simpler methods that need to solve a (sparse) linear system. Therefore, our algorithms are considerably simpler.

(ii) Rather than directly compute subnetworks, network reconciliation algorithms modify the score of each gene in the network based on their interactions. The modified scores can be smoothly incorporated into a number of systems biology analyses, including active subnetwork discovery.

(iii) Moreover, by computing a new ranking of all genes we retain the possibility of discovering multiple pathways related to the disease of interest.

Other authors have proposed using PageRank [95], one of the network reconciliation algorithms described here, to integrate gene expression data with protein interaction networks [69, 92, 136]. PINTA [92] uses the algorithms we have described but evaluates them in a more restricted biological context. They check if knocked-out genes are ranked highly by the algorithms in gene expression data collected following the knock-out. Two applications have used PageRank to study doxorubicin dosage response [69] and to select diagnostic genes in cancers [53, 136]. Our approach extends these previous works in the following ways:

(i) We list three well-motivated criteria to evaluate the algorithms. We demonstrate the dependence of these evaluations on the input parameter q, thereby suggesting suitable values of the parameter.

(ii) We use the reconciliation algorithms to study a diverse set of human diseases. We show that the algorithms can retain differences between diseases while also taking the network structure into account.

(iii) Using a set of seven brain diseases, we demonstrate that these algorithms can reveal genes involved in a relevant biological function, even when those genes were not differentially expressed.

4.3 Methods

Let $G(V, E)$ denote an undirected protein interaction graph in an organism, where V is the set of nodes and E is the set of undirected edges, in which each edge (u, v) represents an interaction between genes $u, v \in V$. We denote the weight of an edge (u, v) by $w_{uv} > 0$. The larger the weight of an interaction, the larger is our belief that u and v indeed physically interact in the cell. Let N_u denote the set of neighbors of node u in G and let $d_u = \sum_{v \in N_u} w_{uv}$ denote the *weighted degree* of u.

Given some biological condition, let $s(v) : V \to \mathbb{R}^+$ be a function that maps genes in V to a non-negative real number representing their degree of perturbation in the contrast

between the condition and an appropriate control (e.g., brain cells from patients diagnosed with Alzheimer's disease and healthy brain cells). The larger the value of $s(v)$, the more we consider the gene to be perturbed in the disease compared to the control. We compute $s(v)$ as the negative absolute value of the base 10 logarithm of the gene's p-value. We normalize the $s(v)$ values so that they sum to 1. These represent the starting values for each node in V. Note that $s(v)$ represents the degree of perturbation of gene v but does not record whether the gene is up- or down-regulated.

Interacting genes often participate in the same protein complex, are members of the same biological pathway, or are controlled by the same transcription factor. Consequently, interacting gene pairs commonly display similar expression profiles. The fundamental intuition underlying our approach is that if two genes u and v are connected by a highly-weighted interaction in G, then $s(u)$ and $s(v)$ should maintain similar values. Furthermore, the larger the value of w_{uv}, the closer $s(u)$ and $s(v)$ should be. This assumption enables the approaches presented here to elucidate highly-relevant genes that may be missed by differential gene expression studies alone. For example, genes within the same complex or pathway may not be individually perturbed to a significant extent, but we may be able to exploit the interactions among them to recognize that the complex or pathway is perturbed as a whole. Guided by this intuition, we propose to compute a value $p(v)$ between 0 and 1 for every node $v \in V$. We want the value of $p(v)$ to simultaneously balance two potentially conflicting constraints:

(i) $p(v)$ remains close to v's initial value $s(v)$.

(ii) $p(v)$ is similar to $p(u)$ for every neighbor $u \in N_v$.

We describe four different methods for computing $p(v)$: Vanilla Algorithm (V), PageRank (PR), GeneMANIA (GM), and Heat Kernel (HK). These methods were developed previously in machine learning and information retrieval [23, 95, 146, 147] and have proven widely applicable in computational biology [42, 53, 68, 69, 85, 92, 130, 132, 136]. Each of these algorithms appears in the literature under different names; we select the most recognizable names from the literature.

For the first three methods, we describe an energy function over the graph G that, when minimized, addresses the two constraints listed above. We then describe an iterative algo-

rithm for each method that efficiently minimizes the energy function and provably converges to the optimum theoretical solution. We are unable to formulate a similar energy function for the fourth method, Heat Kernel. However, we describe a well-known approximation to the discrete heat kernel equation. This approximation yields a computationally efficient iterative solution similar to those used for the other three methods.

4.3.1 Vanilla Algorithm

The Vanilla algorithm seeks to minimize following energy function:

$$\mathcal{E}_{\mathrm{V}} = q \sum_{v \in V} \big(p_{\mathrm{V}}(v) - s(v)\big)^2 + (1-q) \sum_{(u,v) \in E} w_{uv} \big(p_{\mathrm{V}}(v) - p_{\mathrm{V}}(u)\big)^2$$

When we compute the values of $p_{\mathrm{V}}(v)$ that minimize $\mathcal{E}_{\mathrm{V}}$, the first sum ensures that $p_{\mathrm{V}}(v)$ remains close to $s(v)$ for each node v in G, while the second sum ensures that $p_{\mathrm{V}}(v)$ remains close to $p_{\mathrm{V}}(u)$ for every $u \in N_v$. The parameter q, for $0 < q \leq 1$, balances the contribution of each sum to the energy function. Since $\mathcal{E}_{\mathrm{V}}$ is a quadratic function of the $p_{\mathrm{V}}(v)$ values, we can minimize it by setting its partial derivative with respect to each $p_{\mathrm{V}}(v)$ to 0, obtaining the following linear system:

$$p_{\mathrm{V}}(v) = \frac{q s(v) + (1-q) \sum_{u \in N_v} w_{uv} p_{\mathrm{V}}(u)}{q + (1-q) d_v}$$

We compute $p_{\mathrm{V}}(v)$ using an iterative algorithm on G. We initialize the value at node v to $s(v)$. At each iteration $i > 0$, the value at node v is the sum of $\frac{qs(v)}{q+(1-q)d_v}$ (a quantity that depends solely on the initial value) and a contribution $(1-q)\frac{w_{uv}p(u)}{q+(1-q)d_v}$ from each neighbor $u \in N_v$. If we use $p_{\mathrm{V},i}(v)$ to denote the value of node v at iteration i, we can write the following recurrence for $p_{\mathrm{V},i}(v)$:

$$p_{\mathrm{V},i}(v) = \begin{cases} s(v) & \text{if } i = 0, \\ \frac{qs(v)}{q+(1-q)d_v} + & \\ \frac{1-q}{q+(1-q)d_v} \sum_{u \in N_v} w_{uv} p_{\mathrm{V},i-1}(u) & \text{if } i > 0. \end{cases} \tag{4.1}$$

As i tends to infinity, for each node v, $p_{\mathrm{V},i}(v)$ converges to $p_{\mathrm{V}}(v)$, as we prove below.

Proof of convergence of the iterative solution. The proof of convergence of this iterative algorithm is well-known [32]. We include it here for the sake of completeness. We find it convenient to use vector and matrix notation. Let $\mathbf{p_i}$ and $\mathbf{s}$ be $n = |V| \times 1$ column vectors, where $\mathbf{p_i}$ contains the $p_{\mathrm{V},i}(v)$ values and $\mathbf{s}$ contains the quantities $\frac{s(v)}{q+(1-q)d_v}$ for all the nodes $v \in V$. In addition, let W be an $n \times n$ matrix. Every edge $(u, v) \in E$ results in two non-zero entries in W: $W_{uv} = \frac{w_{uv}}{q+(1-q)d_u}$ and $W_{vu} = \frac{w_{uv}}{q+(1-q)d_v}$. All other entries in W are 0. Note that when $0 < q \leq 1$, the sum of the values in every row of W is strictly less than $1/(1-q)$. Let r denote the maximum of these row sums; note that $r(1-q) < 1$.

Now, by combining equation (4.1) for all nodes $v \in V$, we obtain

$$\mathbf{p_i} = q\mathbf{s} + (1-q)W\mathbf{p_{i-1}}$$

By induction, we can prove that

$$\mathbf{p_i} = q\left(\sum_{k=0}^{i-1}\big((1-q)W\big)^k\right)\mathbf{s} + \big((1-q)W\big)^i\mathbf{s} \tag{4.2}$$

For the sake of convenience, we use A to denote the matrix given by $(1-q)W$. The crux of the proof of convergence lies in showing that the values in the matrix A^i tend to 0 as $i \to \infty$. We prove this fact next, by considering the sum of the values in row u in the matrix A^i. This sum is

$$\begin{aligned}
\sum_{v\in V} A^i_{uv} &= \sum_{v\in V}\sum_{x\in V} A^{i-1}_{ux}A_{xv} \\
&= \sum_{x\in V} A^{i-1}_{ux}\sum_{v\in V} A_{xv} \\
&\leq r(1-q)\sum_{x\in V} A^{i-1}_{ux}, \\
&\leq r^i(1-q)^i, \text{ by induction.}
\end{aligned}$$

The penultimate inequality follows from the fact that

$$\sum_{v \in V} A_{xv} = (1-q) \sum_{v \in V} W_{xv} \leq (1-q)r.$$

Since $0 < r(1-q) < 1$, in the limit, the sum of the values in row u in the matrix A^i is 0. Since W is a non-negative matrix, so is A^i. Therefore, in the limit, we conclude each element of A^i converges to 0. Therefore, the sum $\sum_{k=0}^{i-1} A^k$ converges to $(I-A)^{-1}$ in the limit. Substituting this result into equation (4.2), we see that the limiting value of $\mathbf{p_i}$ is given by

$$\mathbf{p} = \lim_{i \to \infty} \mathbf{p_i} = q\big(I - (1-q)W\big)^{-1}\mathbf{s}. \tag{4.3}$$

4.3.2 PageRank

In the formulation of $\mathcal{E}_{\mathrm{V}}$, the contribution of a node is proportional to its weighted degree. Therefore, nodes with high weighted degree may have an unduly large influence on the solution. The PageRank approach [53, 69, 95, 136] accounts for the effect of the weighted degree of each node by minimizing a slightly different energy function on G:

$$\mathcal{E}_{\mathrm{PR}} = q \sum_{v \in V} \frac{\big(p_{\mathrm{PR}}(v) - s(v)\big)^2}{d_v} + (1-q) \sum_{(u,v) \in E} w_{uv} \left(\frac{p_{\mathrm{PR}}(u)}{d_u} - \frac{p_{\mathrm{PR}}(v)}{d_v} \right)^2.$$

In the second sum, we divide each occurrence of $p_{\mathrm{PR}}(v)$ by d_v to adjust for the weighted degree of node v. As before, the parameter q serves to balance the conflicting constraints represented by each of the two sums in $\mathcal{E}_{\mathrm{PR}}$. Since $\mathcal{E}_{\mathrm{PR}}$ is a quadratic function of the $p_{\mathrm{PR}}(v)$ values, we minimize it by setting its partial derivative with respect to each $p_{\mathrm{PR}}(v)$, $v \in V$ to 0, obtaining the following linear system:

$$p_{\mathrm{PR}}(v) = qs(v) + (1-q) \sum_{u \in N_v} \frac{w_{uv}}{d_u} p_{\mathrm{PR}}(u).$$

As in the case of Vanilla, we can compute the quantities $p_{\mathrm{PR}}(v)$ by an iterative algorithm on the graph G. During each iteration, the value of each node v is the sum of $qs(v)$ and a contribution $(1-q)\frac{w_{uv}p(u)}{d_u}$ from each of its neighbors $u \in N_v$. Let $p_{\mathrm{PR},i}(v)$ denote the value

at node v in iteration i of the PR algorithm. We then have the following recurrence for $p_{\mathrm{PR},i}(v)$:

$$p_{\mathrm{PR},i}(v) = \begin{cases} s(v) & \text{if } i = 0, \\ qs(v) + (1-q)\sum_{u \in N_v} \frac{w_{uv}}{d_u} p_{\mathrm{PR},i-1}(u) & \text{if } i > 0. \end{cases}$$

As i tends to infinity, $p_{\mathrm{PR},i}(v)$ converges to $p_{\mathrm{PR}}(v)$ for each node v. The proof of convergence is very similar to the proof for V algorithm, except that in the case of PR, we define $\mathbf{p_i}$ to contain the $p_{\mathrm{PR},i}(v)$ values. Define $\mathbf{s}$ to contain the $s(v)$ values for each node $v \in V$. Finally, define W to be an $n \times n$ matrix where $W_{uv} = \frac{w_{uv}}{d_u}$ and $W_{vu} = \frac{w_{uv}}{d_v}$ for every edge $(u, v) \in E$. Under these assumptions, we get the same equations as in the proof for the Vanilla method, and we naturally prove the same result from Equation 4.3.

$$\mathbf{p}_{\mathrm{PR}} = \lim_{i \to \infty} \mathbf{p}_{\mathrm{PR}}^{(\mathbf{i})} = q\big(I - (1-q)W\big)^{-1}\mathbf{s}.$$

4.3.3 GeneMANIA

The GeneMANIA method [85, 130, 146] is motivated by similar concern from the PageRank method that nodes with high weighted degree may have disproportionately large effect on the final node values. Therefore, GeneMANIA seeks to minimize the following energy function on G:

$$\mathcal{E}_{\mathrm{GM}} = q\sum_{v \in V}\big(p_{\mathrm{GM}}(v) - s(v)\big)^2 + (1-q)\sum_{(u,v)\in E} w_{uv}\left(\frac{p_{\mathrm{GM}}(u)}{\sqrt{d_u}} - \frac{p_{\mathrm{GM}}(v)}{\sqrt{d_v}}\right)^2$$

In the second sum, we divide each occurrence of $p_{\mathrm{GM}}(v)$ by $\sqrt{d_v}$ (compared to dividing by d_v in PR) to adjust for the weighted degree of node v. Again, q balances the contribution from each sum in the energy function. We minimize $\mathcal{E}_{\mathrm{GM}}$ by setting its partial derivative with respect to each $p_{\mathrm{GM}}(v)$ to 0, achieving the following linear system:

$$p_{\mathrm{GM}}(v) = qs(v) + (1-q)\sum_{u \in N_v} \frac{w_{uv}}{\sqrt{d_u d_v}} p_{\mathrm{GM}}(u).$$

As with Vanilla and PageRank, we compute the quantities $p_{\mathrm{PR}}(v)$ using an iterative solu-

tion. During each iteration, the value of each node v is the sum of $qs(v)$ and a contribution $(1-q)\frac{w_{uv}p(u)}{\sqrt{d_u d_v}}$ from each of its neighbors $u \in N_v$. Letting $p_{\text{GM},i}(v)$ denote the value at node v in iteration i of the GM algorithm naturally defines a recurrence as before:

$$p_{\text{GM},i}(v) = \begin{cases} s(v) & \text{if } i = 0, \\ qs(v) + (1-q)\sum_{u \in N_v} \frac{w_{uv}}{\sqrt{d_u d_v}} p_{\text{GM},i-1}(u) & \text{if } i > 0. \end{cases}$$

As i tends to infinity, $p_{\text{GM},i}(v*)$ converges to $p_{\text{GM}}(v)$, using the same proof seen for Vanilla and PageRank by substituting the following definitions. Let $\mathbf{p_i}$ be contain the $p_{\text{GM},i}(v)$ values. Let $\mathbf{s}$ contain the $s(v)$ values for each node $v \in V$. Finally, let W be an $n \times n$ matrix where $W_{vu} = W_{uv} = \frac{w_{uv}}{\sqrt{d_u d_v}}$ for every edge $(u, v) \in E$.

4.3.4 Heat Kernel

The heat kernel of a graph describes the dispersion of heat throughout a network over time. Here, the amount of heat corresponds to the degree of perturbation of each gene, represented by a node in the network, to the disease of interest. The heat kernel is given by the following equation [23, 92, 132]:

$$\mathbf{p} = \sum_{k=0}^{\infty} \frac{(-t)^k}{k!} L^k \mathbf{s} = e^{-tL}\mathbf{s} \tag{4.4}$$

where t parameterizes the rate of heat dispersion, and L describes the edges between nodes in the network. We define $L = I - W$, where I is the identity matrix, and W is a normalized edge weight matrix such that $W_{u,v} = \frac{w_{uv}}{d_u}$. Since directly computing the matrix exponential in Equation 4.4 is computationally expensive, we use the approximation

$$\mathbf{p}_{\text{HK}} = \left(I - \frac{t}{n}L\right)^n \mathbf{s},$$

which converges to Equation 4.4 as $n \to \infty$. We must select n large enough such that the approximation given by $\mathbf{p}_{\text{HK}}$ is sufficiently close to Equation 4.4. We select $n = 100$, as this value performed well for Web graphs [138], which are at least an order of magnitude larger than the biological networks we used. We employ the following algorithm to iteratively

compute $\mathbf{p}_{\text{HK}}$. Let $\mathbf{p}_{\text{HK},0} = \mathbf{s}$, and recursively define

$$\mathbf{p}_{\text{HK},i} = \left(I - \frac{t}{n}L\right)\mathbf{p}_{\text{HK},i-1} = \left(I - \frac{t}{n}L\right)^{i-1}\mathbf{s}.$$

Then we can compute $\mathbf{p}_{\text{HK}} = \mathbf{p}_{\text{HK},n}$ by repeatedly applying the following update function to each node:

$$\begin{aligned}
p_{\text{HK},i}(v) &= p_{\text{HK},i-1}(v) - \frac{t}{n}\sum_{u} L_{u,v}p_{\text{HK},i-1}(u) \\
&= p_{\text{HK},i-1}(v) - \frac{t}{n}p_{\text{HK},i-1}(v) - \frac{t}{n}\sum_{u\neq v} L_{u,v}p_{\text{HK},i-1}(u) \\
&= p_{\text{HK},i-1}(v) - \frac{t}{n}p_{\text{HK},i-1}(v) + \frac{t}{n}\sum_{u\in N_v} W_{u,v}p_{\text{HK},i-1}(u) \\
&= \left(1 - \frac{t}{n}\right)p_{\text{HK},i-1}(v) + \frac{t}{n}\sum_{u\in N_v} W_{u,v}p_{\text{HK},i-1}(u),
\end{aligned}$$

where the penultimate equality arises by substituting L with $I - W$.

4.4 Datasets

Gene expression data. We used publicly-available gene expression data for 54 different human diseases and for the corresponding normal tissues [120]. See Table 4.1 for a complete list of the 54 diseases used in this study. For each disease and its corresponding control, we applied Linear Models for Microarray Data (LIMMA) [114] to the DNA microarray data to compute a t-statistic and a p-value indicating the statistical significance of the differential expression of each gene in that disease, when compared to the corresponding control. We used the negative base 10 logarithm of each gene's p-value as the initial value for each gene in the network, and we normalized these values so they summed to 1. We considered alternatively computing mutual information between each gene and the sample phenotype labels, but we were concerned that the small number of samples may not yield reliable mutual information values. Nonetheless, our methods can be readily applied to mutual information values, and we plan to investigate this extension in future research.

Actinic Keratosis	Idiopathic Thrombocytopenic Purpura
Acute Myeloid Leukemia	Ischemic Cardiomyopathy
Adenocarcinoma of Esophagus	Lung Transplant Rejection
Adenocarcinoma of Lung	Malaria
Alzheimers Disease	Malignant Melanoma
Anaplasmosis	Malignant Neoplasm of Hypopharynx
Bipolar Disorder	Malignant Neoplasm of Prostate
Chronic Obstructive Lung Disease	Malignant Pleural Mesothelioma
Chronic Polyarticular Juvenile Rheumatoid Arthritis	Malignant Tumor of Colon
Chronic Progressive Ophthalmoplegia	Mixed Hyperlipidemia
Clear Cell Carcinoma of Kidney	Myelodysplastic Syndrome
Complex Dental Cavity	Nephroblastoma
Congestive Cardiomyopathy	Obesity
Crohns Disease	Papillary Thyroid Carcinoma
Cystic Fibrosis	Polycystic Ovary Syndrome
Dermatomyositis	Progeria Syndrome
Diabetic Nephropathy	Rett Syndrome
Dilated Cardiomyopathy Secondary to Viral Myocarditis	Rheumatoid Arthritis
Duchenne Muscular Dystrophy	Rotavirus Infection of Children
Endometriosis	Sarcoidosis
Essential Thrombocythemia	Schizophrenia
Glaucoma	Scott Syndrome
Glioblastoma	Senescence
Hereditary Gingival Fibromatosis	Squamous Cell Carcinoma of Lung
Human Immunodefiency Virus Encephalitis	Squamous Cell Carcinoma of Mouth
Huntingtons Disease	Urothelial Carcinoma
Idiopathic Pulmonary Fibrosis	Uterine Leiomyoma

Table 4.1: The list of 54 human diseases used to evaluate network reconciliation [120].

Interaction network. We used the human MiMI network [122] as the underlying protein interaction network in our analysis. MiMI merges interaction data from over 20 interaction databases. We discarded self-edges and repeated edges from the interaction network, resulting in a network comprised of 11,074 proteins and 77,952 interactions. Although MiMI is a very comprehensive resource, a number of interactions included in it were detected by error-prone high throughput experiments.

Typically, there is little information regarding the confidence of interactions discovered experimentally via high- or low-throughput technologies. As a result, many network-based methods operate on unweighted molecular interaction networks [101, 130, 136] even though these methods may be applicable to weighted networks. However, we anticipate that estimating the quality of each interaction will improve the results of network-based algorithms. A number of approaches have emerged in the literature to estimate the confidence of molecular interactions. For example, predicted interaction networks assign a prediction score to each interaction that is often directly applied as an edge weight [68, 92]. Alternatively, binding affinities between transcription factors and their target binding sites may offer an appropriate estimate of interaction confidence [42] when working with transcriptional regulatory networks. Similarly, weighting edges by the number of mutations observed on a pair of interacting genes [130] has proven appropriate for detecting mutated pathways in cancer. Nevertheless, there is no standard approach for estimating the confidence of an interaction, and implemented approaches are often problem-specific.

We desired an edge weighting method that used purely topological measures of reliability, since we incorporated gene expression data into these methods and used functional annotations to interpret the results. We estimated the reliability of each interaction by its FS-weight [22]. Let $G(V, E)$ denote the interaction network, where V is the set of nodes and E is the set of edges in the network. For each edge $(u, v) \in E$, define the weight of (u, v) as

$$w_{uv} = \left(\frac{2|\hat{N}_u \cap \hat{N}_v|}{|\hat{N}_u - \hat{N}_v| + 2|\hat{N}_u \cap \hat{N}_v| + \lambda_{u,v}} \right) \left(\frac{2|\hat{N}_u \cap \hat{N}_v|}{|\hat{N}_v - \hat{N}_u| + 2|\hat{N}_u \cap \hat{N}_v| + \lambda_{v,u}} \right),$$

where $\hat{N}_u = \{u\} \cup N_u$ is the set containing u and its neighbors, and

$$\lambda_{u,v} = \max\left(0, \frac{2|E|}{|V|} - |\hat{N}_u - \hat{N}_v| - |\hat{N}_u \cap \hat{N}_v|\right).$$

Notice that w_{uv} is large when u and v have many neighbors in common and small when they have diverse neighborhoods. Thus, we assign higher confidence to a pair of nodes that share many common interactors in the interaction network.

Functional enrichment. We annotated the genes in our network with 3272 gene sets from MSigDB version 3.0 category C2 [118], 1703 CORUM complexes [107], 223 NCI PID curated pathways [109], and 75 NetPath pathways [57]. We performed all tests for functional enrichment using Model-based Gene Set Analysis (MGSA) [10] directly from the R Bioconductor package. We applied MGSA to the top 250 genes ranked only by their differential expression p-values and to the top 250 genes after applying our reconciliation algorithms.

A wide variety of functional enrichment methods are available in the literature [13, 65, 98, 118]. These approaches typically perform a term-by-term analysis, reporting the significance of the relationship between each function and a collection of genes being studied. The disadvantage of these approaches is that they typically return long lists of significantly enriched functions, from which the user must determine which are the most relevant. After applying FuncAssociate [13], GSEA [118], and PAGE [65] on our datasets, we found it difficult to distinguish top-ranking functions from one another because they annotated similar collections of genes.

However, MGSA [10] simultaneously evaluates all gene sets using a Bayesian approach that integrates overlap between gene sets into the enrichment analysis. MGSA attempts to compute a non-overlapping set of pathways that annotate the study set. MGSA computes a posterior probability for each pathway that reflects how well the pathway overlaps with the study set while not overlapping with other pathways with higher posterior probability. We performed all tests for functional enrichment using MGSA. MGSA allows ranges to be set for two primary parameters α and β. Parameter α controls the fraction of unknown false positive genes, while β controls the fraction of unknown false negatives. We set an upper

limit of the α and β parameters to 0.3 and 0.5, respectively. Thus, less than 30% of the genes annotated by enriched functions are not in the study set (i.e., top 250 genes from the given ranking), and less than half of the genes from the study set are not annotated by any enriched function. All other parameters were left to their default settings.

4.5 Results

We divide our results into six parts. First, in Section 4.5.1, to investigate the effect of the parameter q on these algorithms we analyze the contribution from the network versus the contribution from the expression data to the energy functions that define Vanilla, PageRank, and GeneMANIA. The remaining five subsections of these results address the desirable properties listed in the chapter background. In Section 4.5.2, we examine topological properties of the final gene rankings given by each algorithm, providing insight on how these approaches address the first property. In Section 4.5.3, we simultaneously address the second and third properties by analyzing how well the gene rankings for seven diseases recover the genes in canonical pathways for those diseases. In Section 4.5.4, we discuss similarities between the gene rankings produced by each algorithm across all diseases. Our main concern in these evaluations is the extent to which disease-specific signals are not masked by network-based effects, directly addressing the second desirable property. In Section 4.5.5, we perform functional enrichment tests on the top-ranking genes reported by our reconciliation methods. This analysis further reinforces our conclusions from the previous section and addresses the final desirable property that top-ranking genes should demonstrate functional coherence. Lastly, in Section 4.5.6 we investigate the insulin-mediated glucose transport pathway in several diseases related to the brain. We demonstrate that network reconciliation identifies disease-related proteins from this pathway that are missed by differential expression analysis.

We performed this analysis over a wide range of values for the input parameter q; we report results for $q \in \{0.1, 0.3, 0.5, 0.7, 1\}$. Note that the Heat Kernel is parameterized by $t > 0$. We used the transformation $q = 2^{-t}$ to determine values that covered a reasonable range of possible values for t. This transformation has the additional benefit that q tends to 0 (respectively, 1) as t tends to ∞ (respectively, 0). Since large t increases the dispersion

of heat through the network, the interpretation of q remains the same for all algorithms: large q gives more weight to the starting values, while small q emphasizes network topology.

4.5.1 Balanced energy function contributions

Vanilla, PageRank, and GeneMANIA each minimize an energy function that contains two components: contribution from the gene expression data, denoted by the first sum (over all nodes), and the contribution from the network, denoted by the second sum (over all edges). We use the phrases "node sum" and "edge sum" to informally denote each of these quantities. The parameter q in each energy function serves to balance the weight of the two sums, where high q gives more importance to the gene expression values and low q favors network topology. Our concern is that setting q too small or large may hide signal from the expression values or interaction data, respectively.

We ran Vanilla, PageRank, and GeneMANIA on the MiMI interaction network using gene expression data from 54 different human diseases described in Section 4.4. We applied each algorithm to the 54 diseases using a range of values for the input parameter q and analyzed the total contribution of the node sum versus the edge sum to their corresponding energy function as we varied q. Figure 4.1 plots the negative logarithm of the ratio of the node sum to the edge sum. The variance of this ratio is small for all algorithms across the 54 diseases, indicating that the results are robust to variation in the initial node values (even at high values of q). PageRank and GeneMANIA exhibit balanced energy function contributions from the node sum and the edge sum when $q = 0.5$, while Vanilla balances the two sums only when $q \approx 0.9$. While these observations do not directly shed light on the three desirable properties outlined in the chapter background, we find these results useful in understanding the role of the parameter q in each algorithm. Specifically, this analysis demonstrates that the setting for q should be interpreted separately for each algorithm.

4.5.2 Network coherence

The first desirable property of reconciliation algorithms seeks to identify coherent network structures among top-ranking genes. Each algorithm ranks disease-related genes highly by

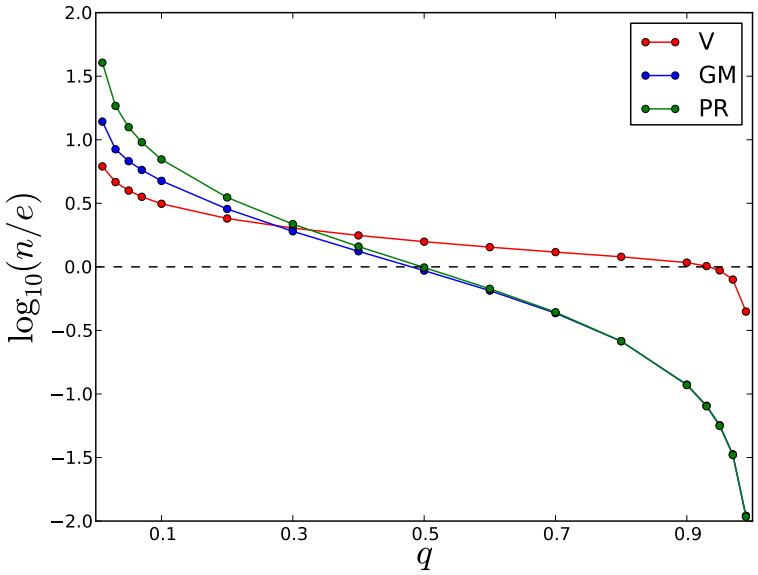


Figure 4.1: Contributions from the node sum (n) and edge sum (e) to the energy functions for Vanilla (V), PageRank (PR), and GeneMANIA (GM) as q varies. Each point represents the average across all 54 diseases.

taking into account both expression and interaction data. However, such rankings may prove difficult to interpret if the top-ranking genes are sparsely connected, inducing many small, connected components. Indeed, a lack of connectivity among such components may conceal the mechanisms by which the disease-related genes interact. We computed the number of connected components induced in the MiMI network by the top k genes reported by each of the four reconciliation algorithms for $0 \leq k \leq 250$.

Figure 4.2 illustrates the change in the number of connected components induced by the top k genes reported by each algorithm for different values of the input parameter q. Each point indicates the average number of connected components across all 54 diseases. The *initial* lines represent the connected components induced by the rankings given solely by the expression data (i.e, $q = 1$); these lines serve as a common reference across the four subplots. Surprisingly, for Vanilla the connectivity among top-ranking genes decreases when more emphasis is placed on the network. Indeed, a decrease in q results in many more connected components among top-ranking genes. Since top-ranking genes reported by Vanilla tend to be less connected as the network is given higher emphasis, we conclude that the Vanilla algorithm performs poorly with respect to the network connectivity property. However, PageRank, GeneMANIA, and Heat Kernel drastically decrease the number of connected

components as q decreases, indicating high network coherence for these algorithms.

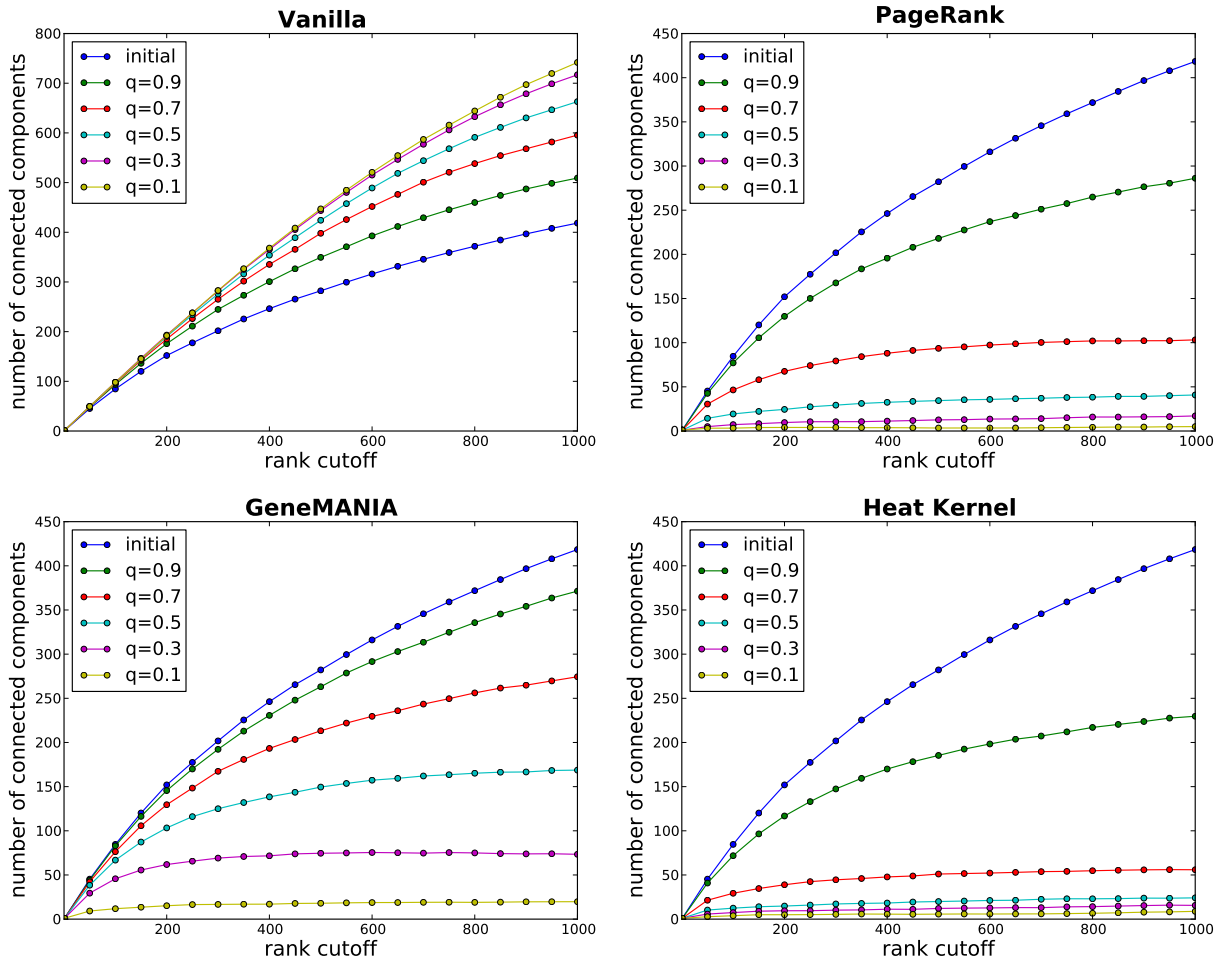


Figure 4.2: Connected components induced by the top-ranking nodes from each algorithm.

Next, we assessed the significance of these connected component counts. We randomly shuffled the gene identifiers in the gene expression data and applied PageRank to the randomized expression data. As in the previous analysis, we computed the number of connected components induced by the top k genes returned by PageRank when we set the parameter q to 1.0, 0.5, and 0.1. We repeated this process 100 times. Figure 4.3 plots the number of connected components induced by the top k genes for 100 random rankings and the ranking given by applying PageRank to the true expression data. Thus, the blue, cyan, and yellow curves in Figure 4.3 correspond to the blue, cyan, and yellow curves in the PageRank subfigure (upper right) of Figure 4.2, and the dashed red lines correspond to the results from 100 random rankings. Figure 4.3 indicates that the number of connected components induced

by the top 1000 genes is lower than the number of connected components given by applying PageRank to any of the 100 random rankings.

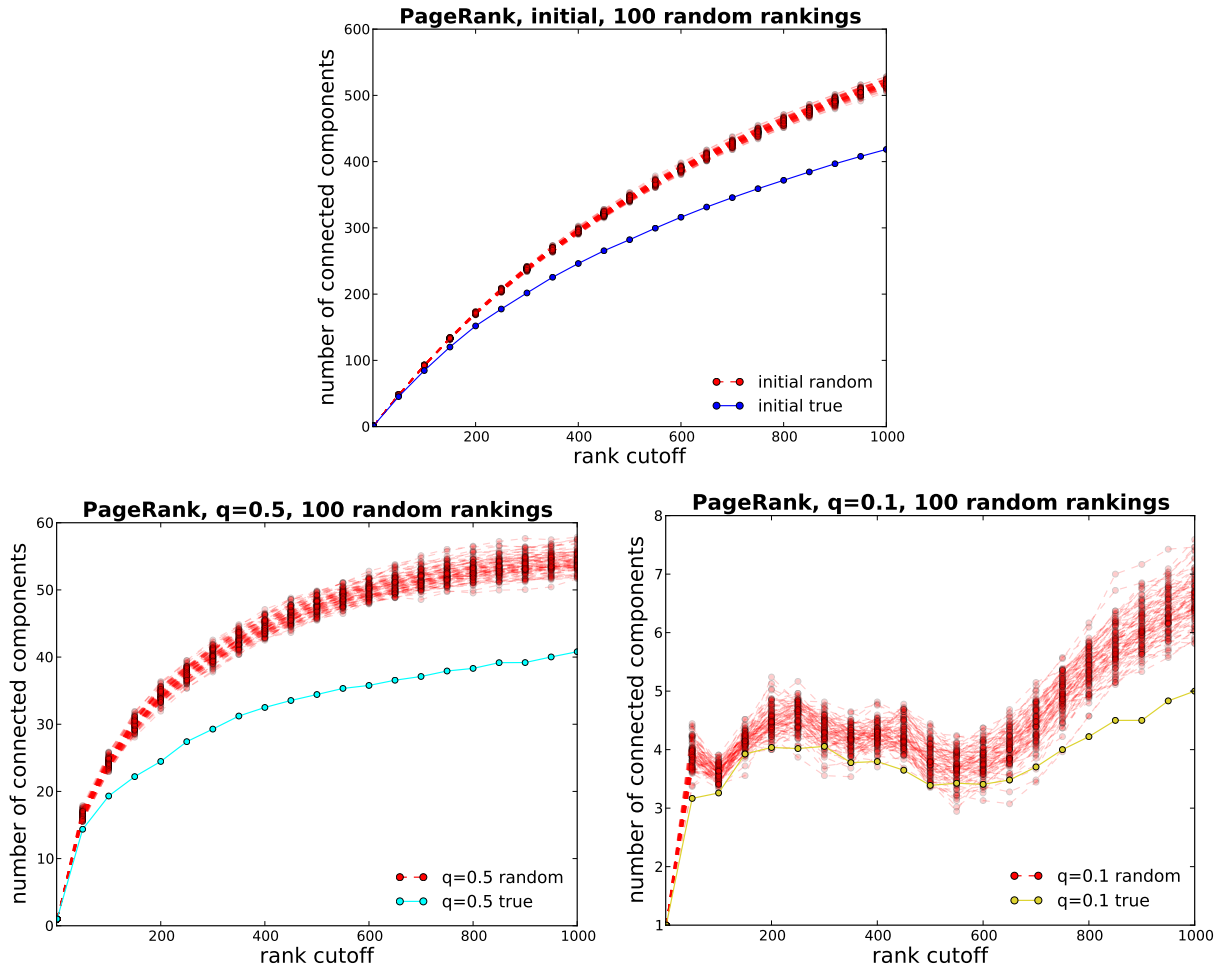


Figure 4.3: Connected components induced by the top-ranking nodes from PageRank applied to 100 randomized gene expression datasets and to the true gene expression data.

4.5.3 Recovering canonical pathways

We assessed the ability of reconciliation algorithms to recover genes involved in the canonical pathway for a disease. Of the 54 diseases in our gene expression data, seven were represented in the Kyoto Encyclopedia of Genes and Genomes (KEGG) [58], which maintains manually-curated pathways of disease mechanisms. We applied the network reconciliation algorithms to each of these seven diseases, namely, malignant melanoma, Huntington's disease, glioblastoma, endometriosis, dilated cardiomyopathy, Alzheimer's disease, and acute

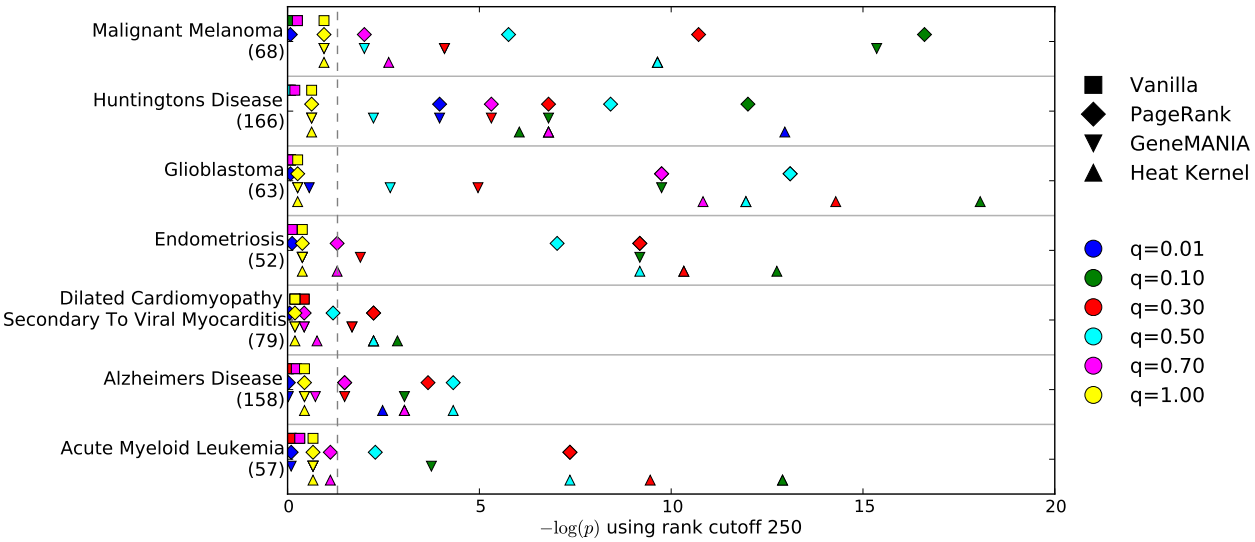


Figure 4.4: Negative logarithms of the hypergeometric p-values indicating the significance of the overlap between the members of seven KEGG pathways and the top 250 genes ranked by each algorithm for the corresponding disease. The number of genes in each KEGG pathway is provided in parentheses. Points to the right of the dashed grey line indicate significant p-values ($p < 0.05$).

myeloid leukemia. We assessed how highly the genes in each disease pathway are ranked by the reconciliation process and by their initial differential expression values.

Figure 4.4 displays the hypergeometric p-values of the overlap between the top 250 genes ranked by each method and the set of genes in the corresponding KEGG pathway. We plot the negative logarithm of each p-value. Points to the right of the dashed grey line indicate significant findings ($p < 0.05$). Figure 4.4 demonstrates that an insignificant number of genes in each of the seven KEGG pathways are among the top ranked genes when considering differential expression p-values alone (yellow points). Additionally, Vanilla does not highly rank a significant number of KEGG pathway genes for any of the seven diseases or for any value of the input parameter q. In contrast, a statistically significant number of KEGG pathway genes appear among the top 250 genes ranked by PageRank, GeneMANIA, and Heat Kernel for every disease and at least one value of q. These findings indicate that reconciling gene expression values with an underlying interaction network provides insights into disease mechanisms that may be missed by expression studies alone.

The value of q that results in the most significant overlap with the KEGG pathway varies considerably depending on the disease and the algorithm. We note that setting $q = 0.5$ for the

PageRank algorithm results in the most significant p-values for two of the three brain diseases with canonical pathways in KEGG (glioblastoma and Alzheimer's disease). Therefore, we set $q = 0.5$ when we perform a focused analysis of brain diseases in Sections 4.5.5 and 4.5.6, though other choices are reasonable. Note that no value of q for Vanilla successfully identifies disease genes from their canonical KEGG pathway.

4.5.4 Similarity between gene rankings

Next, we investigate how each algorithm distinguishes different diseases. Ideally, applying a reconciliation algorithm to transcriptional signatures from two diseases that affect different biological pathways should result in different gene rankings, specifically among the top-ranking genes. For PageRank, GeneMANIA, and Heat Kernel, we examine the difference between the final gene rankings given by each algorithm across all 54 diseases. Since this collection of diseases affects a wide variety of organs, tissues, and cell types, we expect their disease-related genes (and thus the top-ranking genes) to vary considerably.

For each algorithm, we computed the Jaccard index of the top k genes after reconciliation between every pair of diseases for $0 < k \leq 250$. The Jaccard index measures the size of the intersection divided by the size of the union of two sets. Thus, a high Jaccard index between a pair of diseases indicates that the top k genes are highly similar between diseases, while low Jaccard index suggests the algorithm maintains disease specificity in its final rankings. Figure 4.5 illustrates the Jaccard indices for each algorithm. Each point indicates the average Jaccard index across all $\binom{54}{2}$ pairs of diseases. As expected, the Jaccard index increases with a decrease in q for any rank cutoff. Indeed, as q decreases, the network plays a more pronounced role in each algorithm and provides the same signal regardless of the input disease. Thus, to minimize the overlap between the top k genes reported for each pair of diseases, one could simply use the *initial* rankings given solely by the expression data (i.e., $q = 1$). However, this has the obvious drawback that the connectivity of the proteins is completely ignored.

In general, for every value of q the GeneMANIA method results in a much lower Jaccard index than the other algorithms. Selecting $q = 0.1$ is clearly a bad choice, as the Jaccard index of the top k genes between any pair of disease is around 0.7 for PageRank and Heat Kernel. However, setting $q = 0.5$ or higher results in a reasonably low Jaccard index for all

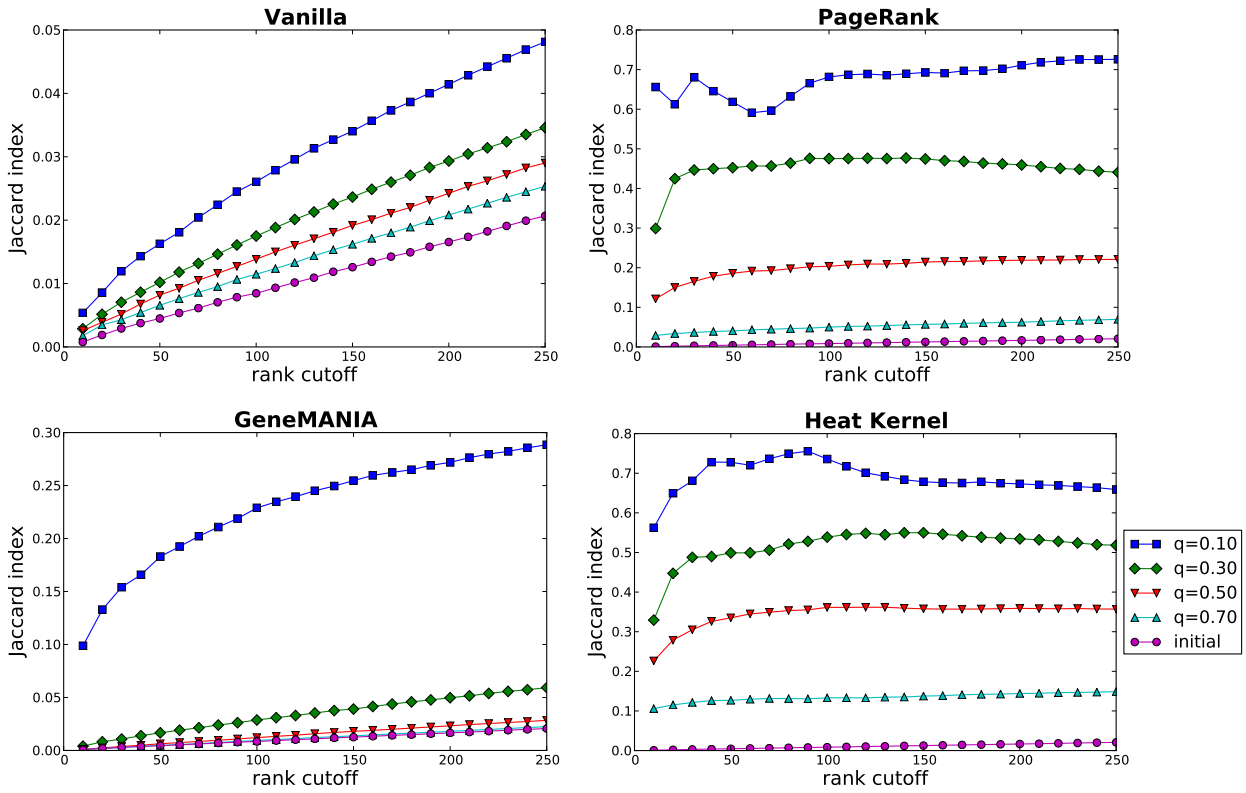


Figure 4.5: The Jaccard index of the top k genes reported by each algorithm for a pair of different diseases. Each point indicates the average Jaccard index of all $\binom{54}{2}$ pairs of diseases using a particular value of q as input to the algorithm.

algorithms. For the functional enrichment analysis presented in the remaining sections, we set $q = 0.5$, as this parameter value jointly addresses the desirable properties for PageRank, GeneMANIA, and Heat Kernel in the analyses presented thus far.

4.5.5 Functional enrichment analysis

Inter-disease functional similarity. We applied functional enrichment tests to further evaluate the gene rankings. Using the gene sets and pathways described in Section 4.4 as protein functional annotations, we applied Model-based Gene Set Analysis (MGSA) to the top 250 genes reported by each reconciliation method on seven diseases that affect the brain: Alzheimer's disease, bipolar disorder, glioblastoma, Huntington's disease, Rett syndrome, schizophrenia, and senescence. Figure 4.6 illustrates the Jaccard index between the top k functions returned by MGSA, $0 < k \leq 50$, for each pair of brain disorders. Note that

we omit the Vanilla algorithm from this analysis, as the top-ranking functions enriched in gene rankings produced by Vanilla were not related to the corresponding disease. Recall from Section 4.5.3 that rankings generated by Vanilla performed poorly at recovering genes from the canonical KEGG pathway for each disease; thus, we expected that highly enriched functions given by the Vanilla gene rankings would be unrelated to the disease. While the Jaccard indices are not as low as those for the MGSA results applied to the *initial* gene rankings given by the expression data, there is a remarkably low overlap between the top functions reported for each pair of diseases. This result supports the findings in Section 4.5.4 that reconciled gene rankings maintain disease specificity. We demonstrate that enriched functions are also relevant to their corresponding diseases in Section 4.5.6.

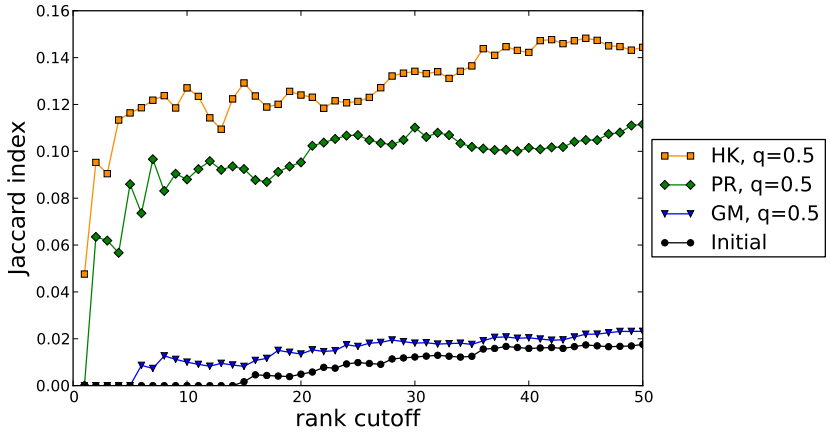


Figure 4.6: Each point denotes the average Jaccard index between the top-ranking functions according to MGSA for all $\binom{7}{2}$ pairs of brain disorders. Note that lower Jaccard index indicates that MGSA identifies dissimilar functions for each pair of diseases.

Inter-algorithm functional similarity. We also investigated the similarity between the functional results of different algorithms. In Figure 4.7, we show the average Jaccard index between the top k functions returned by MGSA for a pair of reconciliation algorithms applied to a single disease. We plot the average across the seven brain disorders. The Jaccard indices are highest for the pair of algorithms PageRank and Heat Kernel at around 0.3 for the first 50 functions. The small index for any pair of algorithms suggests that each algorithm probes a different space of functional annotations for the same disease. We find this to be a particularly striking finding. Since PageRank and Heat Kernel showed high similarity (Jaccard index 0.7)

between the top 250 *genes* reported by each algorithm on the same disease, we expected high similarity in gene rankings to translate into similar functional enrichment results. We plan to further explore this finding in future studies.

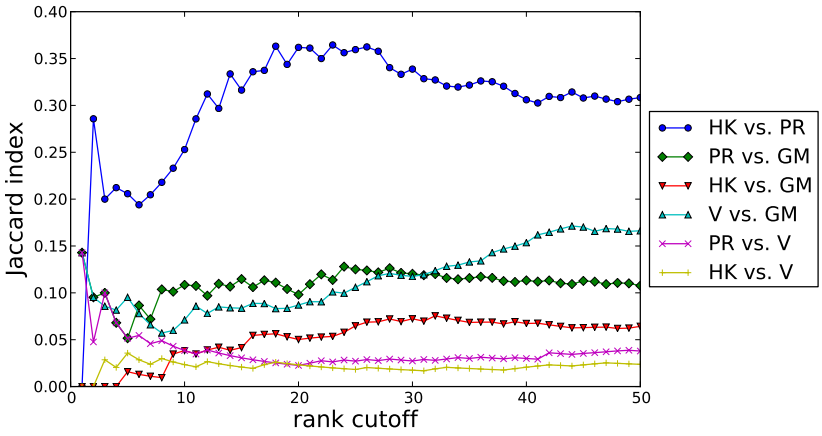


Figure 4.7: Each point denotes the average Jaccard index between the top-ranking functions for a pair of algorithms across all seven brain disorders.

4.5.6 Insulin-mediated glucose transport in the brain

Finally, we further investigated how reconciliation algorithms addressed our fourth desirable property that genes involved in biological functions related to the disease of interest should be ranked highly. We were particularly interested in the effect of reconciliation algorithms on low-ranking genes (i.e., genes with insignificant differential expression between disease and control) annotated by such functions. Ideally, we hoped to see the values of genes involved in biological functions related to the disease were re-ranked highly through the reconciliation process. For this analysis we focused on the NCI PID pathway *insulin-mediated glucose transport*. We selected this pathway because dysregulation of insulin-mediated glucose uptake has been previously implicated in patients diagnosed with various neurodegenerative disorders, including Alzheimer's disease (AD) and Huntington's disease (HD) [28]. After applying PageRank reconciliation, this function appears in the top 22 most enriched functions for four of the seven brain disorders (Table 4.2), including AD and HD.

Alzheimer's disease. Elderly patients diagnosed with AD show impaired glucose tolerance, often erroneously attributed to poor exercise and diet. However, reduced insulin-

Disease	Before PR		After PR	
Alzheimer's Disease	0.0000	(3290)	**0.8839**	**(2)**
Bipolar Disorder	0.0000	(1763)	0.0030	(431)
Glioblastoma	0.0000	(1963)	**0.8340**	**(4)**
Huntington's Disease	0.0060	(130)	**0.3804**	**(22)**
Rett Syndrome	0.0000	(3244)	**0.6333**	**(9)**
Schizophrenia	0.0000	(3115)	0.0000	(3459)
Senescence	0.0000	(3384)	0.0067	(265)

Table 4.2: Enrichment of the NCI pathway *insulin-mediated glucose transport* in the top 250 genes before and after applying PageRank to the expression profiles of seven brain disorders. The values are the posterior probabilities reported by MGSA, where higher value indicates a higher probability that the pathway is enriched. In parentheses, we report the rank of the pathway among 5273 gene sets.

mediated glucose uptake is observed in early stage AD patients whose physical activity and dietary patterns do not differ from healthy adults [28]. Thus, dysregulation of this pathway (i.e., enrichment in disease versus control) may provide an early indication of AD. For each of the seven brain diseases we studied, Table 4.2 reports the MGSA enrichment posterior probabilities for the NCI PID pathway *insulin-mediated glucose transport* in the top 250 proteins before and after applying the PR reconciliation method. This pathway was not enriched in the top 250 proteins ranked only by the differential expression of the corresponding genes. However, PR re-ranked proteins from the insulin-mediated glucose transport pathway highly, drastically increasing the enrichment of the pathway and moving this function from rank 3290 before reconciliation to 2 after applying PR. Thus, we identified a pathway whose role is highly-relevant to AD but is missed using standard functional enrichment methods when *only* the gene expression data is utilized. By integrating the expression profile with a network of protein interactions, we were able to highlight this pathway by re-ranking relevant proteins whose corresponding genes are not significantly differentially expressed.

Huntington's disease. Figure 4.8(a) illustrates the subnetwork induced by proteins in the insulin-mediated glucose transport pathway. At the core of this pathway are seven proteins from the 14-3-3 protein family. The 14-3-3 proteins play a major role in cellular signal transduction, and they are known to appear abundantly throughout the brain. These proteins can bind to a wide variety of other human proteins, altering features of the target

protein such as subcellular localization, functional activity, and phosphorylation state [31]. Figure 4.8(a) demonstrates that most of the genes encoding the 14-3-3 proteins are not significantly differentially expressed in HD versus healthy samples; none of the seven genes appear in the top 250 significantly differentially expressed genes. Table 4.3 reports the ranks of the 14-3-3 proteins before and after reconciliation, with the first protein appearing only at rank 500 before reconciliation. However, Figure 4.8(c) and Table 4.3 demonstrate that the ranks of all seven 14-3-3 proteins increased drastically (along with several additional members of the pathway) after applying PageRank network reconciliation.

The 14-3-3 proteins were re-ranked highly by PageRank because many nearby genes in the network were significantly differentially expressed, and their value propagated through the dense subnetwork of 14-3-3 proteins during the reconciliation process. Notice that the insulin-mediated glucose transport pathway was ranked highly in HD before and after applying PR (Table 4.2). However, 12 proteins from this pathway appeared in the top 250 proteins after reconciliation compared to just five before reconciliation. Thus, in this case reconciliation did not identify a novel pathway related to the disease (as we discovered with AD), but reconciliation successfully re-ranked highly relevant proteins that would be missed using differential gene expression alone. Furthermore, the role of this pathway in HD may be easier to interpret since the involvement of the 14-3-3 proteins is highlighted by reconciliation but missed otherwise.

Protein	**Before PR**	**After PR**
YWHAQ	500	210
YWHAE	874	79
YWHAZ	1286	112
YWHAS	6188	231
YWHAB	6380	48
YWHAG	7594	32
YWHAH	8433	351

Table 4.3: Ranks of 14-3-3 family proteins before and after applying PageRank to the HD expression data.

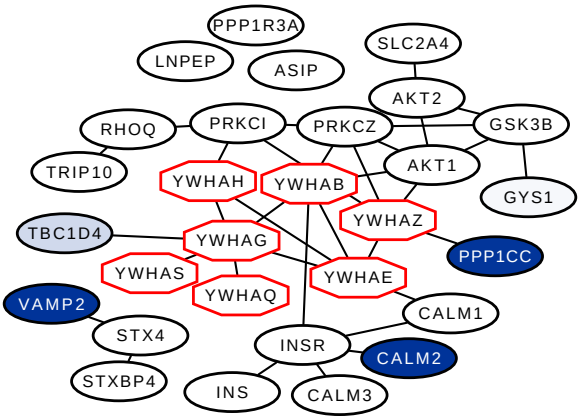
High degree proteins. One concern with our methodology is that the reconciliation approaches may overemphasize proteins with many neighbors in the network. Indeed, the

reconciliation methods may be more likely to propagate value to nodes with many neighbors, and our analysis of AD and HD may be biased, as we have already mentioned that the 14-3-3 proteins are highly promiscuous. However, notice that the insulin-mediated glucose transport pathway is not found to be enriched in bipolar disorder, schizophrenia, or senescence after reconciliation (Table 4.2). In fact, this pathway is even less enriched in schizophrenia after reconciliation, demonstrating that the enrichment results are not biased by such high degree nodes.

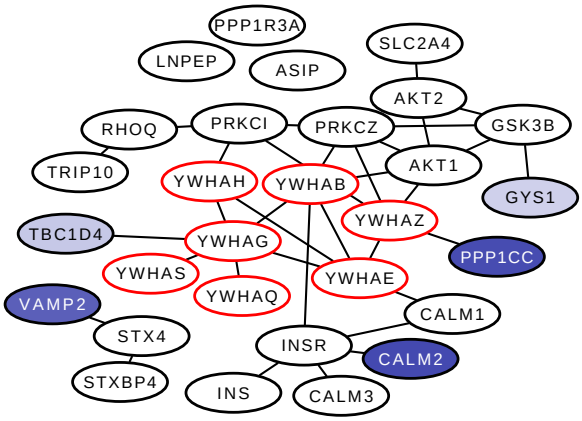
4.6 Discussion

We described four approaches to integrate case-control gene expression data with molecular interaction networks. These methods actively modify gene expression measurements to match the constraints imposed by the edges in the network, while controlling the deviation of the modified values from their original settings. We enumerated three desirable biological properties that this class of algorithms should address. These properties aim to balance input from gene expression data with an underlying interaction network while maintaining that the returned gene rankings are specific to the condition being studied. Addressing any one of these properties alone may be trivial. For example, to address the first property that top-ranking genes should participate in coherent network structures, one could simply return a list of genes such that each gene in the list is connected to one of its predecessors in the list. Thus, any cutoff in the gene ranking induces a single connected component. However, addressing all three properties simultaneously is more difficult and warranted further investigation. We analyzed each algorithm using differential gene expression data from a variety of human diseases that interrogate vastly different organs and tissues. Ultimately, this work attests to the wide applicability of reconciliation algorithms and suggests reasonable values of their input parameters to address the three desirable properties. We demonstrated that i) PageRank, GeneMANIA, and Heat Kernel always outperform Vanilla with respect to our primary motivating properties and ii) applying any of these three network reconciliation algorithms then analyzing the resulting gene ranks yields more interpretable results than analyzing the ranking of significantly differentially expressed genes alone.

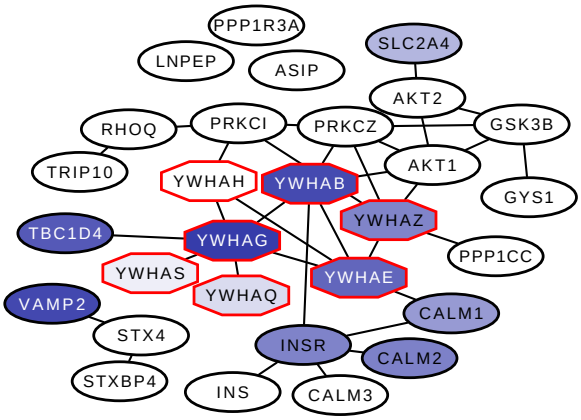
In the future, we plan to consider other biologically-relevant formulations of the energy function. An important extension is to situations that violate our assumption that two linked genes should be similarly perturbed in a condition. For instance, a transcription factor may be connected to a target gene that it down-regulates. In this situation, it seems appropriate to ensure that the transcription factor and its target have distinct values. More generally, the expression of a gene may have a very complex dependence on those of its interactors. When such relationships are known, it will be useful to incorporate them into our formulation. Lastly, our methods utilize the significance of a gene's differential expression while ignoring whether the gene is up- or down-regulated. Promising extensions may incorporate the direction of regulation as well as the mechanism of regulatory interactions.



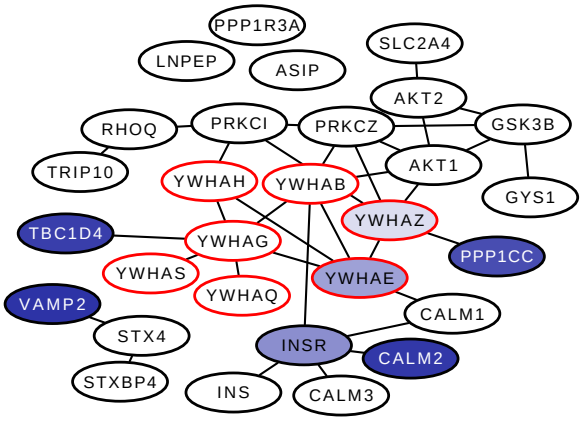
(a) Initial insulin-mediated glucose transport network.



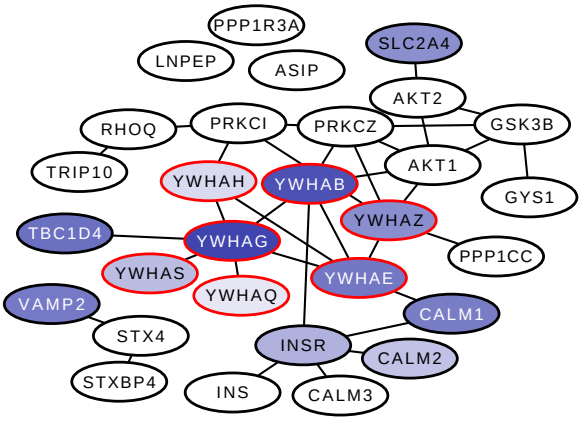
(b) After Vanilla.



(c) After PageRank.



(d) After GeneMANIA.



(e) After Heat Kernel.

Figure 4.8: A comparison of the subnetwork induced by genes involved in the NCI pathway *insulin-mediated glucose transport* with nodes weighted by (a) differential expression p-values from patients diagnosed with Huntington's disease, and after applying (b) Vanilla, (c) PageRank, (d) GeneMANIA, and (e) Heat Kernel. Blue nodes indicate genes ranked in the top 250, and darker nodes indicate higher ranking. Nodes with a red outline indicate genes in the 14-3-3 family of proteins.

Chapter 5

Top-Down Network Analysis to Drive Bottom-Up Process Modeling

Christopher L. Poirel, Richard R. Rodrigues, Katherine C. Chen, John J. Tyson, and T. M. Murali. Top-Down Network Analysis to Drive Bottom-Up Modeling of Physiological Processes. *Journal of Computational Biology*, to appear May 2013.

5.1 Background

Complex networks of interacting genes, mRNAs, proteins and metabolites control the physiological processes of a living cell. These interactions integrate signals from the cell's environment and from its own internal state in order to control appropriate responses in terms of gene expression, cell growth and division, movement, and programmed cell death. A major goal of molecular systems biology is to understand how complex networks of interacting genes and proteins control these basic aspects of cell physiology. Computational cell biologists have succeeded in constructing detailed, mechanistic, and predictive models of fundamental physiological processes such as phage λ development [5], bacterial chemotaxis [4], metabolic regulation in *E. coli* [26], and the yeast cell cycle [4, 5, 21, 26]. We consider these to be primary examples of "bottom-up" models. At the other end of the spectrum, computational "top-down" analyses of high-throughput information about molecular interactions have been used to construct high-level maps of cellular process, e.g., sets of genes that display concerted

activity across a diverse set of cellular conditions [9, 56]. The response network algorithms surveyed in Chapter 2 are fundamental examples of top-down analytical methodologies.

While both top-down and bottom-up approaches have identified interesting and useful patterns of molecular activity, they have intrinsic and complementary limitations. Although networks built using top-down approaches are not limited by the number of components in the network, it is difficult or computationally infeasible to attribute mathematical models to the entire network. Top-down analysis can be used to identify genes and proteins that play roles in various processes within the cell, but it cannot provide detailed predictions of cell phenotypes, e.g., the size of a yeast cell during a specific phase of the cell cycle. Conversely, bottom-up models can be used to make such detailed predictions about cellular response under a variety of internal and external stresses, e.g., after gene knockouts or after over-expressing various proteins. These predictions can then suggest biological experiments for verification. However, constructing mechanistic models is a painstaking, multi-year undertaking that involves careful study of the literature and incremental improvements to existing models.

Since the strengths and weaknesses of these two approaches are largely complementary, we are developing a principled methodology that links high-level, network topologies with detailed dynamic models of cellular control systems. Connections found by the top-down data-analysis approach can suggest novel mechanistic hypotheses that may have escaped the attention of the most attentive molecular biologists. Bottom-up dynamic modeling can then explore the properties and predictions of these hypotheses with depth and rigor unachievable by educated guesswork. Bottom-up dynamic modeling can predict the phenotypes of new designer mutants that could be used to discriminate between alternative network hypotheses, thereby providing a path toward experimental validation. In this work, we address the first step, i.e., analyzing molecular interactomes to suggest extensions to dynamic models.

We use the cell division cycle in *S. cerevisiae* as our motivating example. Cell cycle regulation in budding yeast is an ideal test bed for this methodology, because (i) extensive repositories of genome, transcriptome, proteome and interactome data on yeast are publicly available, (ii) there exists—as a starting point—a detailed mechanistic model of the yeast cell cycle [21], and (iii) flexible and powerful genetic tools are available for budding yeast.

Exploiting these advantages, Chen *et al.* proposed a well-known dynamic model of this process [21]; we refer to this model as CHEN2004 henceforth. CHEN2004 is a collection of biochemical reactions, modeled as ordinary differential equations, that describe protein synthesis and degradation, complex formation, and regulatory activity for 27 genes known to be involved in regulating the yeast cell cycle. Figure 5.1 provides a wiring diagram for CHEN2004, illustrating the major proteins involved in the model and their interacting partners. These reactions are modeled by a set of ordinary differential equations that describe the rate of change of each species (i.e., a protein or protein complex from the model) as a function of the quantities of other species in the model. By solving these ODEs numerically, Chen *et al.* simulated the changing quantities of every species in the model as a wild-type cell progresses through the cell cycle. To refine and test the model, Chen *et al.* then tried to simulate the unique physiological characteristics of 131 mutant strains of budding yeast. In each simulation, changes were made to the "wild-type" parameter set to reflect the genetic makeup of the mutant. For example, if the *CDC20* gene is deleted, then the rate constant for synthesis of Cdc20 protein is set to 0, and the model must reproduce the phenotype of the *cdc20*Δ deletion strain ("inviable, blocked in metaphase"). CHEN2004 faithfully reproduces the phenotypes of 120 mutants in a collection of 131 mutant strains of budding yeast.

Despite its success, the CHEN2004 model is incomplete. Several proteins that are widely recognized as influential in cell cycle progression were not included as species in the model. We developed LINKER to address this issue: given a set of "source" proteins, a set of "target" proteins involved in cell cycle regulation (from CHEN2004), and a yeast interactome, we sought highly relevant and interpretable paths that connect the sources to the target proteins through edges in the interactome. Before developing our methods, we articulate the following important requirements:

(i) Many of the interactions in CHEN2004 are regulatory, e.g., transcription factors govern the synthesis of proteins and kinases phosphorylate their targets. Therefore, the method should be applicable to a directed network.

(ii) We sought formulations of the problem that are computationally tractable (e.g., not NP-complete), to avoid the need for approximation algorithms [143] or heuristics [7].

Ideally, we desired problems that are amenable to polynomial-time algorithms.

(iii) Analysis of the interactome may generate not only potential pathways that are novel and worth pursuing but also networks that , to the trained eye of a yeast molecular geneticist, are trivial or implausible for reasons not evident from the databases. Therefore, modelers and molecular biologists may desire to ask the methods to expand the computed subnetworks. Accordingly, we wanted methods that support an easy-to-interpret parameter, upon whose increase the computed subnetworks would expand smoothly.

We summarize related research in Section 5.2. Next, we present LINKER in Section 5.3, starting with a discussion of how we used these requirements to design our approach. In Section 5.4, we describe our input datasets. In Section 5.5, we evaluate LINKER and compare it to related algorithms.

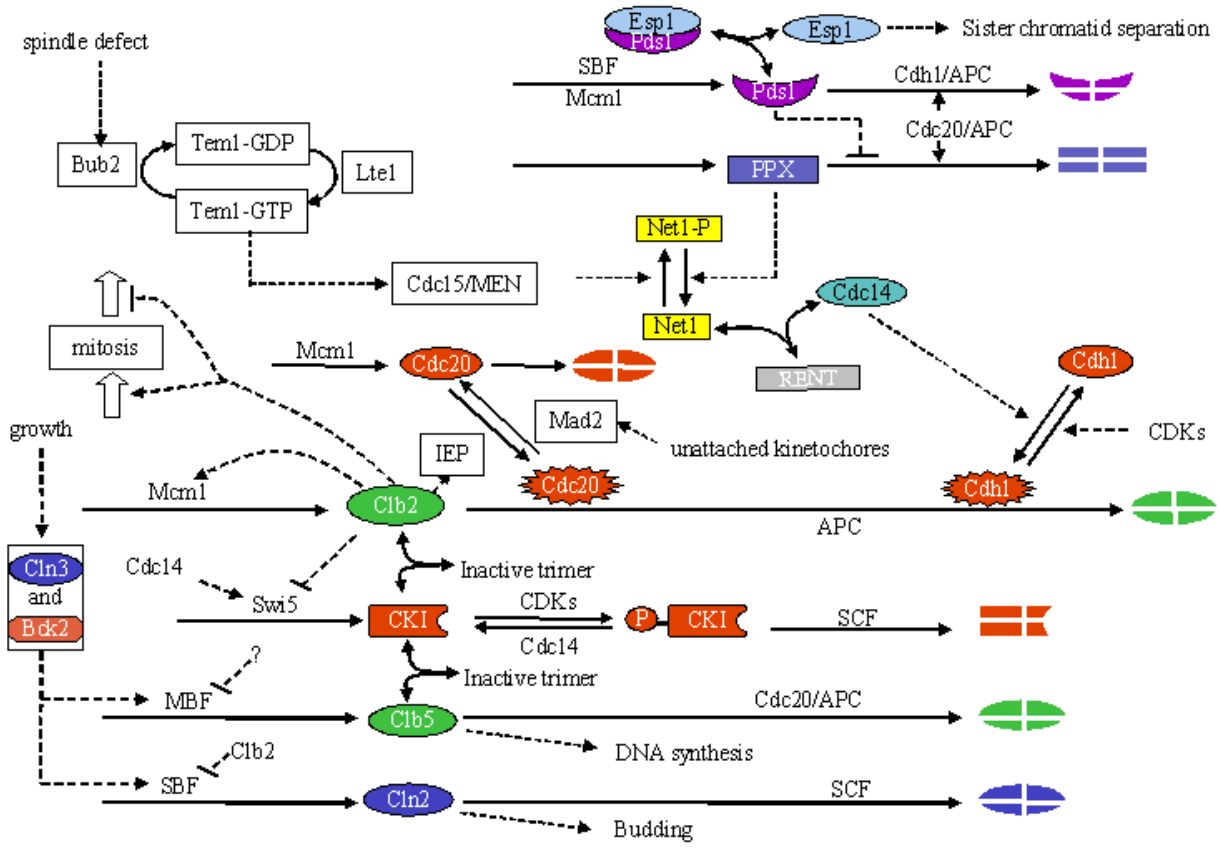


Figure 5.1: Wiring diagram of the yeast cell cycle model by Chen *et al.* [21].

5.2 Related Research

LINKER addresses a problem that belongs to the following class. We are given a weighted, directed interaction network $G = (V, E)$, where V is a set of vertices, E is a set of directed edges, and each edge (u, v) is assigned a weight w_{uv}. Let S be a set of source nodes, and let T be a set of target nodes in the network. The goal is to identify a connected, low weight subnetwork of G that connects the source set to the target set. Many large subnetworks (e.g., G itself) trivially achieve this goal; thus, constraints are typically placed on the structure of the subnetwork in an effort to identify small subnetworks that connect S to T through highly probable (or low cost) edges. Several variations of this problem and related algorithms have emerged recently in systems biology.

ANAT seeks a subnetwork connecting S (which contains a single node in their approach) to T that simultaneously addresses both a local and a global minimization criterion [143]; As before, introduce a dummy source node s connected to each node $v \in S$. The local problem seeks to identify G', a subtree of G, connecting s to every terminal node in T such that the total cost of the path in G' from s to each terminal is minimized. The local problem can be solved in polynomial time using Dijkstra's single-source shortest path algorithm. The global minimization problem seeks a subtree G' such that the sum of the costs of the edges in G' is minimized. This formulation is known as the *minimum Steiner tree problem* and is NP-hard. ANAT accepts as input a parameter that controls the relative importance given to each criterion. ANAT computes a solution that has a provable approximation guarantee. Thus, the subnetwork inferred by ANAT is not necessarily optimal, but the sum of the costs of the edges in the discovered subnetwork is guaranteed to be less than a small factor of the optimal subnetwork. To our knowledge, ANAT does not support directed networks.

RESPONSENET is a flow-based approach that connects source to target proteins through edges in G [141]. The method introduces an artificial source node s connected to each $v \in S$ and an artificial sink node t connected to each $v \in T$. The cost of an edge c_{uv} defines the maximum flow capacity of edge (u, v). RESPONSENET solves a linear program that maximizes the total flow from the source nodes to the target nodes, subject to an additive penalty for allowing positive flow along edges of low weight. The constraints ensure that (i)

the amount of flow traversing edge (u, v) does not exceed its capacity c_{uv}, (ii) the incoming flow to a node must equal its outgoing flow for all $v \in V$, and (iii) the initial flow leaving s must equal the final flow entering t.

The NP-hard *prize-collecting Steiner tree* (PCST) problem is another relevant formulation. Given a weighted, undirected interaction network $G = (V, E)$ and a positive prize b_v associated with each node $v \in V$, the PCST is a connected subtree G' of G that minimizes the sum of the costs of the edges in G' and the prizes of the nodes that are not in G' over all possible connected subgraphs of G. This formulation allows some nodes to be left out of G' if the cost of including them is large. Systems biologists have used an exact mathematical programming solution to PCST [75] to compute cellular response networks [30], interpret metabolomic data [29], and recover signaling pathways [49]. Since PCST is an NP-hard problem, MsgSteiner offers a heuristic approach that computes a quasi-optimal subtree with a depth that is at most d, a user-defined parameter [7]. Additionally, MsgSteiner expects as an input a node r as a root for the subtree; thus, r is guaranteed to be a node in the solution, and the subtree can be directed from the root to the target nodes.

In Section 5.5.3, we describe how we adapt ResponseNet and MsgSteiner to our application, and we compare Linker to these state of the art algorithms.

5.3 Methods

In our application, the modelers preferred to augment Chen2004 one source at a time; hence, S contained one node. Moreover, it was sufficient for our application to find paths connecting the node in S to *any* node in T. The principles outlined at the end of Section 5.1 guided our development of Linker. Rather than directly compute a low-cost subgraph that connects the source S to the targets in T, we solve the problem in two stages. First, we used a teleporting random walk popularly known as PageRank [95] to rank nodes in G with respect to the node in S. We took inspiration from a number of applications of random-walk-like ideas in systems biology [69, 86, 89, 99, 130, 136]. Second, we used the node visitation probabilities to compute the k most-probable paths from S to T. Both steps can take directed graphs as input, satisfying the first requirement. Both steps can be executed efficiently in

polynomial time, satisfying the second requirement. Finally, our approach meets the third desideratum as follows: by increasing k, we have the flexibility to return additional paths, if so requested by the user. By construction, the k best paths are a subset of the $k+1$ best paths, thereby ensuring that the results vary smoothly with k.

The PageRank Algorithm. Although PageRank is well-known, we describe it for the sake of completeness. In general, PageRank can accept multiple nodes in S. We set the starting probability s_v for each node v as follows: $s_v = 1/|S|$ for $s \in S$ and $s_v = 0$ otherwise. Let w_{uv} be the weight of the directed edge (u, v). We normalize the edge weights such that $\sum_{x \in N_u} w_{ux} = 1$, where N_u is the set of out-neighbors of node u, i.e., $N_u = \{v|(u, v) \in E\}$. We start the process by placing the walker on node u with probability s_u for each node $u \in V$. Now, the walker moves in the network according to the following rules, where $0 \leq q \leq 1$ is a parameter:

Teleport: With probability qs_x, she teleports to any node $x \in V$, including her current node and its neighbors; the total probability of teleporting from u is q since $\sum_{x \in V} s_x = 1$.

Walk: She can move from her current node u to any of u's out-neighbors $v \in N_u$ with probability proportional to $(1-q)w_{uv}$; thus, the total probability of walking to some neighbor of u is $1-q$ since $\sum_{x \in N_u} w_{uv} = 1$.

The parameter q provides control over how often the walker teleports back to one of the start nodes in S.

Visitation probabilities. PageRank naturally defines a transition matrix $\mathcal{U}$ among the nodes in G. Each entry $\mathcal{U}_{uv} = qs_v + (1-q)\frac{w_{uv}}{\sum_{x \in N_u} w_{uv}}$ indicates the probability that a walker at node u will transition to node v; thus, $\mathcal{U}$ is right stochastic. Let U be the weighted, directed graph defined by the adjacency matrix $\mathcal{U}$. If U is strongly connected and aperiodic, i.e., the greatest common divisor of the cycle lengths of U is 1, then PageRank converges to a unique stationary distribution. In this case, we can compute the stationary visitation

probability p_v for each node $v \in V$ by solving the following linear system:

$$p_v = qs_v + (1-q)\sum_{u \in N_v} \frac{w_{uv}}{d_u} p_u$$

We can solve the system by inverting an appropriate matrix. In practice, we used the well-known iterative power method to compute the stationary probabilities. We found that this linear system had a unique solution for all our source nodes.

Discovering highly probable paths. Ultimately, we seek interpretable paths that connect the source node to the target nodes. Accordingly, we search for paths from the node in S to any node in T such that the product of the probabilities of the nodes in the path is maximized. We identify the top k most probable paths, where k is a user-defined parameter, by modifying the network and solving the following problem. Given a network G and two nodes s and t, identify the top k shortest paths from s to t. We modify the network by adding an artificial sink node t with a directed edge (v, t) for each node $v \in T$. We assign a cost to each edge (u, v) in the resulting network as follows:

$$c_{uv} = \begin{cases} -\log(p_v) & \text{if } v \in V \\ 1 & \text{if } v = t. \end{cases}$$

We define the cost of a path as the sum of the costs of the edges in the path; thus, the path with lowest cost is equivalent to the most probable path. We search for the k shortest loopless paths from the node in S to t using Yen's algorithm [142], which runs in $O\left(knm + kn^2 \log(n)\right)$ time, where $n = |V|$ and $m = |E|$.

5.4 Datasets

Interactome. We represented the yeast interactome as a directed network $G(V, E)$, where V is the set of yeast genes or their corresponding products and E is the set of edges, in which each edge (u, v) represents an interaction from u to v. The interactome integrated physical protein-protein interactions from BioGRID [116], transcription factor-target interactions

from YEASTRACT [1], and kinase- and phosphatase-target interactions from KID [112] and Bodenmiller *et al.* [16]. We represented BioGRID interactions as bidirectional edges. We directed each YEASTRACT interaction from the transcription factor to its target gene. We manually partitioned the phosphoproteomic interactions from KID into directed or bidirectional based on the experimental evidence codes, with a directed interaction pointing from the kinase or phosphatase to its target protein. We directed all interactions from Bodenmiller *et al.* similarly. We removed all edges connecting pairs of genes in CHEN2004, and we removed nodes with an unweighted in- or out-degree greater than 1000. The network contained 6556 nodes and 151,993 directed edges.

Edge weights. We assigned a confidence $w_{uv} \in (0, 1]$ to each directed edge $(u, v) \in E$ using a probabilistic approach similar to that of Yeger-Lotem *et al.* [141]. Given a collection of experimental evidence sources that support each interaction, this method estimates the probability that the pair of proteins interact. The approach assigns higher confidence to pairs of interacting proteins that participate in the same biological processes.

Here, we provide a formal discussion of the edge weighting procedure. Given a pair of proteins u and v, let $I \in \{0, 1\}$ be a binary random variable such that $I = 1$ if u and v truly interact and $I = 0$ otherwise. Let $E = [E_1, \ldots, E_n] \in \{0, 1\}^n$ be a vector of binary random variables where $E_i = 1$ if experiment i supports an interaction between u and v and $E_i = 0$ otherwise. To each edge we compute a score c_{uv} representing the confidence that u and v interact given the experimental evidence for this pair as

$$\begin{aligned} c_{uv} &= \Pr(I = 1|E) \\ &= \frac{\Pr(E|I = 1)\Pr(I = 1)}{\Pr(E)} \qquad (5.1) \\ &= \frac{\Pr(E|I = 1)\Pr(I = 1)}{\Pr(E, I = 0) + \Pr(E, I = 1)} \\ &= \frac{\Pr(I = 1)\prod_k \Pr(E_k|I = 1)}{\Pr(I = 0)\prod_k \Pr(E_k|I = 0) + \Pr(I = 1)\prod_k \Pr(E_k|I = 1)}, \qquad (5.2) \end{aligned}$$

where Equation (5.1) is an application of Bayes rule, and Equation (5.2) assumes conditional independence of the experimental evidence types (conditioned on I), i.e., $\Pr(E|I) =$

$\prod_k \Pr(E_k|I)$.

Let P and N be disjoint sets of true positive and true negative pairs, respectively. We constructed the set of gold standard positive protein pairs P as all pairs (u, v) such that both u and v were co-annotated by at least one of the Gene Ontology (GO) biological processes listed by Meyers *et al.* [88]. Expert biologists manually curated this list by selecting GO terms specific enough to be verified experimentally and general enough to be tested using high-throughput experiments. We randomly selected protein pairs that were not co-annotated by any of these biological functions as the set of negative protein pairs N, and we chose $|N| = 10 \cdot |P|$ such pairs. We computed the prior probability of an interaction $P(I)$ as

$$\Pr(I = i) = \begin{cases} \frac{|P|}{|P \cup N|}, & \text{if } i = 1 \\ \frac{|N|}{|P \cup N|}, & \text{if } i = 0. \end{cases}$$

Letting X_k be the set of protein pairs *observed* to interact under experiment k (i.e., the set of edges in the interactome with evidence code k), we computed the probability of an individual experiment E_k conditioned on I as

$$\Pr(E_k = e|I = i) = \begin{cases} \frac{|P \cap X_k|}{|P|}, & \text{if } e = 1, i = 1 \\ \frac{|N \cap X_k|}{|N|}, & \text{if } e = 1, i = 0 \\ \frac{|P \setminus X_k|}{|P|}, & \text{if } e = 0, i = 1 \\ \frac{|N \setminus X_k|}{|N|}, & \text{if } e = 0, i = 0. \end{cases}$$

Tables 5.1, 5.2 and 5.3 report the confidence scores for individual experimental evidence codes from BioGRID, KID, and YEASTRACT, respectively. Table 5.3 additionally reports the confidence scores for the Bodenmiller *et al.* experiment and the "Miscellaneous" category, which is the union of all experimental evidences that identified fewer than 25 interactions. The confidence reported for each experiment k was calculated as $\Pr(I = 1|E)$ where $E_k = 1$ and $E_j = 0$ for experiment $j \neq k$. Many edges were discovered from multiple experiments, thus when weighting edges in the network, we computed $\Pr(I|E)$ where E is the true vector of experimental evidence codes for the pair of nodes incident on that edge. We computed

BioGRID Experimental Evidence	**Confidence**
BioGRID FRET	0.937574
BioGRID Far Western	0.920805
BioGRID Co-purification	0.892810
BioGRID Co-crystal Structure	0.882684
BioGRID Reconstituted Complex	0.857869
BioGRID Affinity Capture-Western	0.853658
BioGRID Co-localization	0.834424
BioGRID Co-fractionation	0.823699
BioGRID Protein-peptide	0.667185
BioGRID Two-hybrid	0.573224
BioGRID Affinity Capture-MS	0.553885
BioGRID PCA	0.447374
BioGRID Biochemical Activity	0.360544
BioGRID Affinity Capture-RNA	0.261703
BioGRID Protein-RNA	0.134615

Table 5.1: BioGRID experimental evidence confidence scores. The confidence reported for experiment k is calculated as $\Pr(I = 1|E)$ where $E_k = 1$ and $E_j = 0$ for experiment $j \neq k$.

confidence values close to 1 for many interactions. Such edges may have an unduly large influence on our network-based algorithms, thus we imposed a cap of 0.75 on all edge confidence scores, similar to the approach of Yeger-Lotem *et al.* [141].

Source proteins and targets. For our initial test of Linker, we chose 12 source proteins (Cdc5, Cdc55, Far1, Hsl1, Hsl7, Msn5, Nrm1, Sln1, Ssa1, Stb1, Tpk2, and Ydj1) that were not included in Chen2004 but that are currently being incorporated into extensions of the model, led by Chen and Tyson. Hence, we have some expectations about how these proteins should be connected to the Chen2004 model. We wanted to determine whether Linker could reproduce our own expert knowledge about the control system and if it could suggest new interactions of which we may have been unaware. We included all 27 proteins in Chen2004 as targets.

5.5 Results

We divide our results into four parts: i) an evaluation of how similar PageRank node visitation probabilities are for different queries, ii) how PageRank ranks genes whose deletions modify cell size, iii) a comparison of Linker to ResponseNet and MsgSteiner, and iv) a case study of the network computed by Linker for the protein kinase Cdc5. We note

KID Experimental Evidence	**Confidence**
KID In vivo site-directed mutagenesis in substrate showing same biological consequence as the kinase delete	0.909526
KID LTP Co-localization	0.897122
KID In vivo phosphorylation site mapping using phospho-specific antibodies (Western blot) or by phospho-peptide mapping	0.897121
KID Phosphorylation or kinase-dependent change in localization	0.865948
KID Phosphorylation reduced or absent in kinase mutant (Phospho-shifts, Western blot using Phospho-specific antibody)	0.850282
KID HTP in vitro phosphorylation	0.788615
KID In vitro phosphorylation site mapping (Mass Spec, Phospho-specific antibodies by Western, in vitro site-directed mutagenesis)	0.785787
KID Reconstituted complex	0.784842
KID Physical interaction by Two-hybrid or PCA	0.763252
KID LTP in vitro kinase assays	0.719305
KID Co-Immunoprecipitation / Co-purification	0.688028
KID Reduction in phospho-peptide in vivo by mass-spec	0.642800
KID Yeast 2-Hybrid studies or PCA assay	0.630194
KID Co-Immunoprecipitation by Mass Spec	0.416234
KID Localized to same subcellular compartment	0.360907
KID Protein Chip data for in vitro phosphorylated substrates	0.222044

Table 5.2: KID experimental evidence confidence scores. The confidence reported for experiment k is calculated as $\Pr(I = 1|E)$ where $E_k = 1$ and $E_j = 0$ for experiment $j \neq k$.

YEASTRACT Experimental Evidence	**Confidence**
YEASTRACT indirect: s1 nuclease protection assays - wild type vs tf mutant	0.900429
YEASTRACT indirect: rt-pcr - wild type vs tf mutant	0.605936
YEASTRACT direct: emsa	0.583261
YEASTRACT indirect: northern blotting - wild type vs tf mutant	0.579494
YEASTRACT direct: dna footprinting	0.575631
YEASTRACT indirect: lacz - wild type vs tf mutant	0.505055
YEASTRACT indirect: gfp - wild type vs tf overexpression	0.425588
YEASTRACT direct: lacz - wild type vs target promoter mutant	0.392438
YEASTRACT indirect: proteomics - wild type vs tf mutant	0.326350
YEASTRACT indirect: microarrays - wild type vs tf mutant	0.197915
YEASTRACT direct: chip-on-chip	0.193221
YEASTRACT direct: chip	0.176664
YEASTRACT indirect: microarrays wt vs tf mutant	0.119441
Bodenmiller *et al.* [16]	**Confidence**
Bodenmiller phosphorylation	0.232228
"Miscellaneous" Experimental Evidence	**Confidence**
Miscellaneous	0.602762

Table 5.3: YEASTRACT and miscellaneous experimental evidence confidence scores. The confidence reported for experiment k is calculated as $\Pr(I = 1|E)$ where $E_k = 1$ and $E_j = 0$ for experiment $j \neq k$. We additionally report the confidence scores for the Bodenmiller *et al.* interactions and the Miscellaneous collection of interacting pairs.

that LINKER takes less than 20 seconds to preprocess the inputs, execute PAGERANK, and compute the 50 most probable paths.

5.5.1 Similarity of PageRank Results Between Queries

One concern with PAGERANK is that the structure of the network may dominate the influence of the source when we compute the visitation probabilities, i.e., the most visited nodes may be independent of the source, especially for small values of the input parameter q, since a small value of q rarely resets the walk via teleportation back to the source. To assess the tradeoff between network effects and source selection, we ran LINKER using each of the 12 individual source proteins for three values of the input parameter q, namely, 0.1, 0.5, and 0.9.

After each application of PAGERANK, we ranked the nodes in the network in decreasing order of their visitation probabilities. For each value of q, we measured the Jaccard index of the top-ranking k nodes for each pair of source proteins, where k ranged from 20 to 100; we did not evaluate larger values of k since they were likely to be uninteresting for the purpose of expanding CHEN2004. Figure 5.2 illustrates box plots of the distribution of Jaccard indices of the top-ranking genes over the $\binom{12}{2}$ pairs of sources. Note that we expect the Jaccard index between the rankings for any pair of proteins to increase as we increase k, since the Jaccard index must reach 1 for a sufficiently large value of k. We observed that the similarity between the top-ranking proteins for different sources decreased as we increase the parameter q. For large values of q, the random walker frequently restarts her walk at the source protein, and she is biased toward visiting proteins that can be reached in just a few steps from the source. Since the median Jaccard index with $q = 0.5$ was sufficiently small (less than 0.1) for the top 100 nodes, we used this value of q for all subsequent analyses.

5.5.2 Ranks of Cell Size Modifiers

Jorgensen *et al.* [54] performed a comprehensive genome-wide analysis of the effect of approximately 6000 single gene deletions on cell growth. They systematically measured the cell size distribution in each deletion strain and compared it to the size distribution of wild type

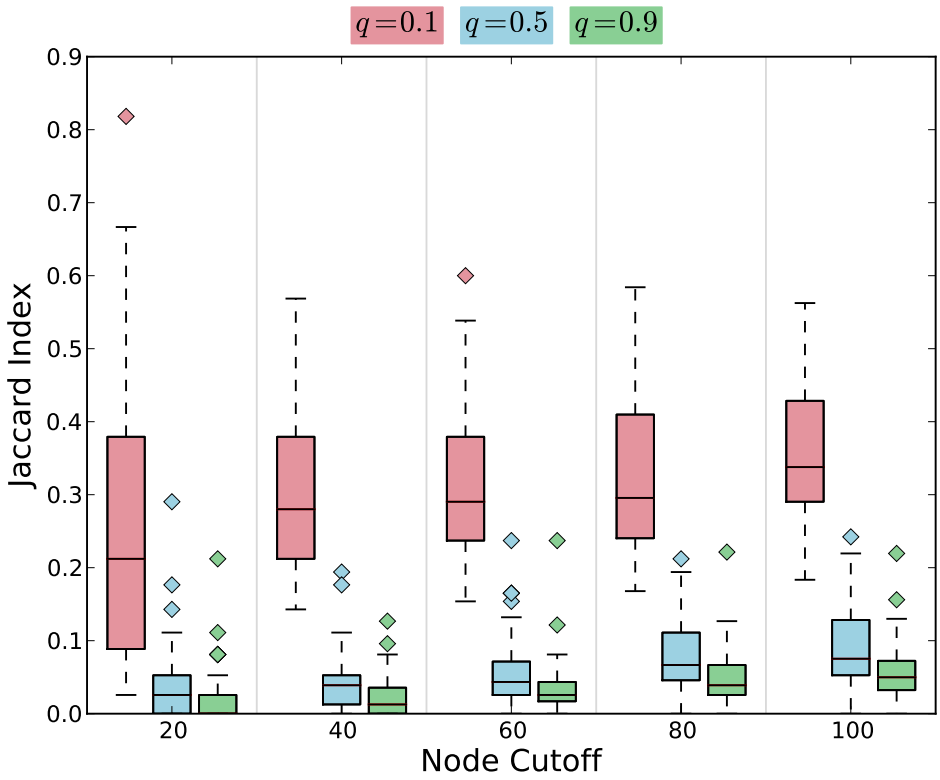


Figure 5.2: The Jaccard index between the top-ranking nodes for each pair of queries. Each box plot illustrates a distribution of all $\binom{12}{2}$ pairs of queries at the specified cutoff using a particular value of q as input to the algorithm.

cells. Jorgensen *et al.* identified 228 and 215 deletion mutants whose cell size distribution is drastically increased or decreased, respectively, compared to wild type.

The cell cycle controls the rate at which the cell grows, including its size at division. Since the source genes that we supplied to PageRank are key players in the cell cycle, we anticipated that PageRank would favor visiting these cell size modifiers during the random walk process. We tested this hypothesis by computing the cumulative distributions of cell size increasers, decreasers, and modifiers (i.e., the union of increasers and decreasers) in the top 100 nodes ranked by PageRank visitation probability. We compared this distribution to the uniform distribution of cell size changers using the Kolmogorov-Smirnov test. Table 5.4 indicates that we observed a statistically significant number of cell size increasers to lie in the 100 most visited nodes. Similarly, for all but one source protein (Sln1), we observed a statistically significant number of cell size modifiers. For 7 of 12 proteins, we observed a significant number of cell size decreasers in the top 100 nodes.

Source	Size increasers	Size decreasers	Size modifiers
CDC5	1.57×10^{-22}	—	6.66×10^{-19}
CDC55	1.57×10^{-22}	5.96×10^{-06}	1.21×10^{-07}
FAR1	8.24×10^{-24}	6.67×10^{-22}	8.24×10^{-24}
HSL1	9.30×10^{-31}	—	3.97×10^{-25}
HSL7	1.12×10^{-20}	—	4.52×10^{-14}
MSN5	2.17×10^{-10}	8.08×10^{-11}	8.08×10^{-11}
NRM1	1.75×10^{-19}	6.67×10^{-22}	4.48×10^{-20}
SLN1	4.52×10^{-14}	—	—
SSA1	5.06×10^{-30}	1.31×10^{-02}	3.64×10^{-23}
STB1	3.70×10^{-12}	8.44×10^{-26}	4.48×10^{-20}
TPK2	8.44×10^{-26}	—	8.44×10^{-26}
YDJ1	8.66×10^{-34}	8.08×10^{-11}	3.28×10^{-17}

Table 5.4: p-values of the Kolmogorov-Smirnov test comparing the observed distribution of cell size increasers, decreasers, or modifiers (i.e., increasers and decreasers) in the top 100 nodes ranked by PageRank visitation probabilities to the uniform distribution. Dashes indicate that fewer increasers, decreasers, or modifiers were observed than expected.

5.5.3 Comparing Linker to ResponseNet, MsgSteiner and KSP

We compared Linker to ResponseNet [141] and MsgSteiner [7], two state of the art algorithms that compute connections between source and target proteins. We also compared Linker to KSP, an approach that directly identifies the k most confident paths from the source to any target, i.e., without initially applying PageRank.

We provided the interactome, each of the 12 source proteins individually, and the set of target cell cycle proteins as inputs to ResponseNet. Ten of the computed subnetworks consisted of only direct interactions between the source and a subset of the target proteins. The number of targets directly connected to the source varied between two and eight, depending on the source. Only two source proteins, Sln1 and Hsl7, yielded non-trivial networks that included three and four internal nodes (i.e., non-source or non-target) in the solution; those two sources had no direct connections with the targets in the interactome. To address our third requirement (where the modeler requests more links from the source to target), we varied ResponseNet's parameter γ that controls the size of the output network. We tested a range of values for γ, where $\gamma \in \{5, 10, 15, 20, 25\}$, as suggested by the authors. However, modifying this parameter had no effect on the output networks for a specific source. While ResponseNet discovered potentially relevant connections from the source to a subset of

the target proteins, we were unable to elicit larger subnetworks from the method.

We applied MsgSteiner to the same inputs, seeking for each query a low-cost connected tree that includes at least one of the target proteins and is rooted at the source. We set the prize of each target protein to 1 and all other nodes to 0. MsgSteiner accepts two input parameters: d specifies the maximum depth of the tree, and λ scales the value of the prizes on each node. We tested values of d ranging from 3 to 6, and three values of λ, namely, 0.125, 0.1875, and 0.25. Increasing either d or λ yielded increasingly larger trees in most, but not in all, cases. However, this variation did not fully address our third requirement: determining reasonable values for d and λ was tedious, as these values depended on the distribution of edge weights. For example, when we used Cdc5 as the source and fixed $d = 4$, varying λ from 0.125 to 0.1875 increased the size of the output tree from 9 to 23 proteins, and the number of collected target nodes increased from 8 to 18. Moreover, gradually increasing d or λ did not grow the computed networks smoothly. Many nodes and edges that were present in the solution for one value of λ disappeared upon increasing λ. Thus, we found that the parameters are difficult to interpret in the context of our application and that MsgSteiner does not readily facilitate a modeler's potential request to expand computed subnetworks.

Lastly, we applied KSP, a naïve version of Linker that excludes the random walk and seeks the k most highly confident paths from the source to any target, where k is an input parameter and the confidence of a path is the product of weights of the edges in the path. The parameter k clearly offers the same advantages for KSP as it does for Linker; increasing k by one identifies the single most probable path from source to target that has not yet been computed.

We assessed the functional coherence of the subnetworks computed by each algorithm through functional enrichment. We applied Model-based Gene Set Analysis (MGSA) [10] to the union of the set of internal nodes (i.e., non-source and non-target) in the subnetworks discovered by each algorithm for the 12 source proteins. We computed the enrichment of Gene Ontology biological processes in each set of internal nodes, where we defined significantly enriched functions as those with an estimated posterior probability greater than 0.5. Table 5.5 lists the enriched functions identified for each invocation of the algorithms. Note that the table only reports an algorithm and its corresponding parameters if they yielded at least one

enriched functions. We used all combinations of parameters mentioned previously for each algorithm. MGSA identified zero enriched functions in the internal nodes from ResponseNet for all parameter choices. We only observed one enriched function for MsgSteiner for four parameter combinations; all other combinations of parameters yielded no enriched functions. Note that we tested twelve total parameter combinations for MsgSteiner (four values for d and three values for λ). MGSA identified one, four, and four significantly enriched functions in the internal nodes for KSP, for k equal to 10, 20, and 30, respectively. We discovered enriched functions for all parameter values given to Linker: four, four, and two enriched functions for the top 10, 20, and 30 most probable paths, respectively. The processes enriched in Linker networks included cell shape checkpoint (the presence of this term supports our earlier results on cell size modifiers), regulation of cytokinesis, and cell shape checkpoint. Including the random walk procedure in Linker clearly improved the functional coherence of genes in most probable paths from source nodes to Chen2004 genes compared to KSP and MsgSteiner.

5.5.4 Interpreting Linker Subnetworks

In Chen2004 the representation of the mitotic exit network (MEN) was incomplete, neglecting the role of Cdc5 (an essential kinase) in the release of Cdc14 (an essential phosphatase) from its association with Net1 in the nucleolus. Since 2004 we have revised the model to include Cdc5 and its targets of phosphorylation (such as Net1, Cdh1, and Bub2/Bfa1). Hence, Cdc5 serves as a useful test case for Linker. Asking for the 10 shortest paths, we obtained the graph shown by solid nodes and edges in Figure 5.3.

Linker succeeded in discovering the major connections between Cdc5 and cell cycle regulatory proteins. The connections to Bub2, Lte1, Cdc14 and Cdc15 capture the role of Cdc5 in the MEN pathway for Cdc14 release. The links Esp1 and Net1 recover Cdc5's known role in the pathway called FEAR ('Cdc14 early-anaphase release'). The interaction between Cdc5 and Cdh1 plays a well-known role in mitotic exit and re-establishing cells in the G1 phase of the cell cycle.

The interaction between Cdc5 and Tem1 was unexpected and particularly useful to the modelers. Looking at the experimental evidence supporting this link, we found evidence for

Algorithm	Parameters	Enriched Function	Score
LINKER	$k = 10$	SRP-dependent cotranslational protein targeting to membrane, translocation	0.972156
		cell shape checkpoint	0.9186824
		activation of MAPKK activity	0.6717416
		regulation of cytokinesis	0.5640634
	$k = 20$	SRP-dependent cotranslational protein targeting to membrane, translocation	0.9863416
		cell shape checkpoint	0.8668016
		regulation of cytokinesis	0.5174146
		activation of MAPKK activity	0.507788
	$k = 30$	cell shape checkpoint	0.7922704
		regulation of exit from mitosis	0.5509068
KSP	$k = 10$	cell shape checkpoint	0.9843462
	$k = 20$	regulation of exit from mitosis	0.7266564
		SRP-dependent cotranslational protein targeting to membrane, translocation	0.6884412
		regulation of spindle pole body separation	0.5822936
		regulation of fungal-type cell wall organization	0.509839
	$k = 30$	SRP-dependent cotranslational protein targeting to membrane, translocation	0.9721356
		regulation of exit from mitosis	0.7075808
		negative regulation of autophagy	0.5566982
		positive regulation of RNA polymerase II transcriptional preinitiation complex assembly	0.5038384
MSGSTEINER	$\lambda = 0.1875, d = 4$	regulation of spindle pole body separation	0.8683294
	$\lambda = 0.25, d = 3$	cell shape checkpoint	0.8484602
	$\lambda = 0.25, d = 5$	regulation of sequence-specific DNA binding transcription factor activity	0.8618764
	$\lambda = 0.25, d = 5$	regulation of sequence-specific DNA binding transcription factor activity	0.8493982

Table 5.5: GO biological processes enriched in the internal nodes returned by each algorithm. The table reports algorithm and parameter combinations for which at least one function was enriched (i.e., MGSA posterior probability of at least 0.5).

a role for Cdc5 in bringing Cdc15 to the spindle pole body where it is needed to activate Dbf2 (the endpoint of the MEN pathway) [104]. This role of Cdc5, of which we were unaware until LINKER brought it to our attention, resolves a long-standing problem with versions of the model developed since CHEN2004 in simulating the activity of Dbf2 in *bub2*Δ *cdc5*Δ cells. Figure 5.4 illustrates two regulatory control mechanisms of Dbf2 by Cdc5. In our present model, Cdc5 exercises control over Dbf2 only through the Bub2-Bfa1 complex: Cdc5 $\dashv$ Bub2-Bfa1 $\dashv$ Tem1 $\rightarrow$ Dbf2. Hence, the model suggests Dbf2 should be active in *bub2*Δ *cdc5*Δ cells, but experiments suggest that Dbf2 is inactive [70]. However, if Cdc5 has a second role, Cdc5 $\dashv$ Bfa1 $\dashv$ Tem1, then the model explains the inhibition of Dbf2 in the double-deletion

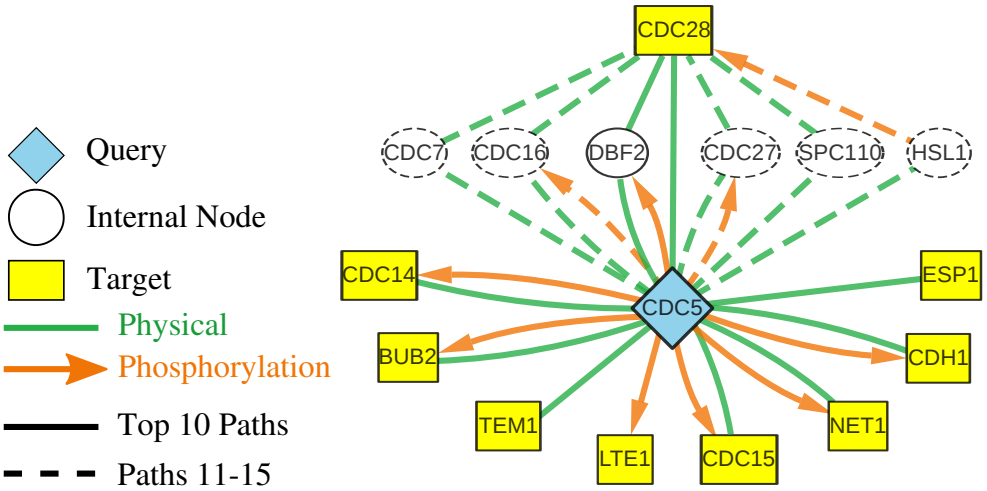


Figure 5.3: The $k = 15$ shortest paths connecting Cdc5 to the cell cycle proteins. Solid nodes and edges indicate those used in the top 10 paths, while dashed nodes and edges denote those used only by the 11^{th} to 15^{th} shortest paths.

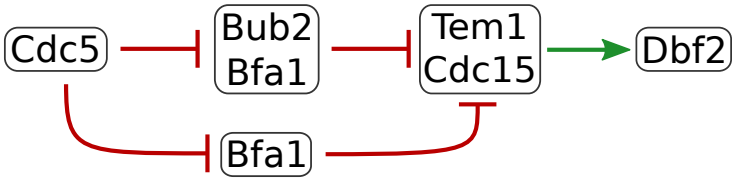


Figure 5.4: Two regulatory control mechanisms of Dbf2 by Cdc5.

mutant. Based on this analysis, we propose the following extensions to Chen2004 that incorporate the second regulatory role of Cdc5 elucidated by Linker (i.e., Cdc5 ⊣ Bfa1 ⊣ Tem1); for consistency, we use the same notation as Chen2004:

$$\frac{d[TEM1_f]}{dt} = \frac{k_{atem} \cdot [LTE1_a] \cdot ([TEM1_T] - [TEM1_f])}{J_{atem} + ([TEM1_T] - [TEM1_f])} - \frac{k_{item} \cdot [BFA1BUB2] \cdot [TEM1_f]}{J_{item} + [TEM1_f]}$$

$$\frac{d[BFA1]}{dt} = \frac{(k_{abfacdc14} \cdot [CDC14] + k_{abfapp2a} \cdot [PP2A]) \cdot ([BFA1_T] - [BFA1])}{J_{abfa} + ([BFA1_T] - [BFA1])} - \frac{k_{ibfacdc5} \cdot [CDC5P] \cdot [BFA1]}{J_{ibfa} + [BFA1]}$$

$$\frac{d[BFA1BUB2]}{dt} = k_{asbfa1bub2} \cdot [BFA1] \cdot [BUB2] - k_{dibfa1bub2} \cdot [BFA1BUB2]$$

$$[MEN] = \frac{[CDC15] \cdot [TEM1_f]}{[CDC15] + [BFA1]}$$

Cdc5 plays additional roles in the DNA damage checkpoint, the morphogenesis checkpoint

and cytokinesis that did not appear in the 10-path graph. The 15-path graph, shown using solid and dashed nodes and edges in Figure 5.3, includes a link between Cdc5 and Hsl1, which is indicative of Cdc5's role in the morphogenesis checkpoint. This part of the network is further elaborated in the 10-path graphs associated with Hsl1 and Hsl7 as source proteins (Figure 5.5). Looking deeper into the Cdc5 graph, we expect to find its connections to the DNA damage checkpoint and to cytokinesis.

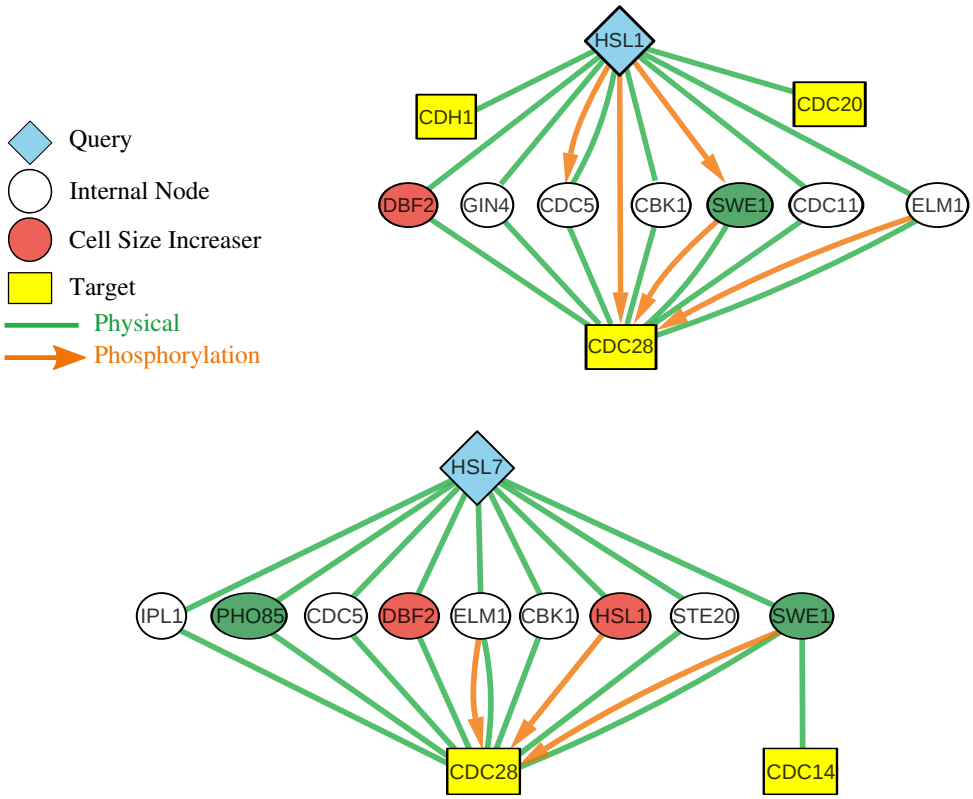


Figure 5.5: The $k = 10$ shortest paths connecting Hsl1 and Hsl7 to the cell cycle proteins.

5.6 Conclusions

In this paper, we have presented LINKER, a method that can identify connections from a given source protein to a set of proteins collectively involved in a particular cellular process. We used LINKER to suggest extensions to CHEN2004, a well-studied dynamic model of the yeast cell cycle. LINKER succeeded in discovering critical connections between Cdc5 and cell cycle regulatory proteins that have been included in the model since 2004. Moreover,

LINKER suggested a link between Cdc5 and Tem1 that resolved a long-standing problem with the model's ability in simulating the activity of Dbf2 in *bub2*Δ *cdc5*Δ cells.

We anticipate that this network-based analysis of the yeast cell cycle will continue to provide useful suggestions of the biological mechanisms by which a query protein affects other proteins in the model. We plan to use this information to guide further development of the existing model and to base subsequent experiments on the improved model. In this manner, we aim to create an integrated framework for hypothesis generation, testing and validation, in order to construct predictive models of complex macromolecular regulatory systems, with the yeast cell cycle as a driving example.

Chapter 6

Automated Reconstruction of Signaling Pathways

Christopher L. Poirel, Anna M. Ritz, Hyunju Kim, Allison Tegge, and T. M. Murali. Automated reconstruction of human signaling pathways. Manuscript in preparation, 2013.

6.1 Background

Cells are constantly bathed in a sea of external signals that initiate internal responses. A primary method of cellular signaling is triggered by the binding of ligands to receptors that initiate protein signaling cascades, activate transcriptional regulators and transcription factors, and culminate in the perturbation of the expression of target genes and subsequent downstream events [2]. Identifying the specific sequence of events and reactions that take place to guide the propagation of these signals is a major focus of systems biology.

Traditionally, signaling pathways are represented by diagrams that illustrate the collection of molecular interactions involved in a particular process. Hand-drawn pathways provide highly intuitive visual representations of signaling mechanisms. However, it is difficult to incorporate such visualizations into a computational framework for analyzing signaling pathways. Over the past two decades, databases have been developed to store the interactions for multiple signaling pathways [57, 60, 81, 96, 109], facilitating their retrieval for computational analyses. These representations are iteratively improved over the years by careful analysis

of the literature and as new biological insight is gained. However, these pathways are still largely built through extensive and time-consuming manual curation. Furthermore, conceptually similar pathways can vary considerably across databases. These drawbacks inspire the development of computational approaches that can automatically reconstruct signaling pathways.

Methods have emerged over the past decade to automatically recover signaling pathways from a much larger background network of molecular interactions (the *interactome*). Steffen *et al.* [117] exhaustively enumerated all short paths in the interactome from every signal receptor to every transcription factor in budding yeast. Scott *et al.* [110] extended this approach by assigning reliability scores to the interactions, and they utilized an algorithmic technique called *color coding* to efficiently identify high-scoring paths. Their modifications to the color coding approach allowed them to search for longer paths and alternative graph structures (e.g., trees). Other methods have utilized the powerful Prize-Collecting Steiner Tree (PCST) problem formulation to integrate proteomic and transcriptomic measurements into a succinct subnetwork [7, 49, 124]. These methods assign appropriate prizes to nodes and costs to edges. They subsequently compute a tree that simultaneously minimizes the total cost of the edges included in the tree and the sum of the prizes of the nodes omitted from the tree. Gitter *et al.* tackled the problem of identifying signaling interactions while simultaneously predicting edge directionality by formalizing the maximum edge orientation problem and developing approximation algorithms to solve their formulation [41]. In the same vein of research, Gitter *et al.* utilized input-output hidden Markov models to integrate time-series expression profiles with a molecular interaction network to reconstruct signaling pathways [40]. A common feature of all these methods was that they computed a subnetwork of the underlying interactome and subsequently demonstrated overlaps of proteins from known signaling pathways with the subnetwork. Most of these methods have been applied to data in budding yeast.

We take a fresh approach to reconstructing signaling pathways by addressing the following challenge: Given the interactome for an organism and the sets of signaling receptors and downstream transcription factors and transcriptional regulators (collectively abbreviated as TRs) in a specific pathway (e.g., the Wnt pathway), recover the signaling interactions for

that pathway from the interactome. We present an algorithm LINKER, that combines a teleporting random walk with k-shortest path computations to compute the most probable paths from the pathway's receptors to its TRs. LINKER proposes a ranked list of interactions involved in the given signaling pathway. We explicitly evaluate the accuracy of the reconstructed pathway by comparing it to the "gold standard" interactions from a database. *We anticipate pathway reconstruction algorithms can complement and even replace parts of the laborious manual pathway curation process by automatically proposing signaling interactions.* We make the following contributions through this work:

i. We apply LINKER to a diverse collection of human signaling pathways; most previous approaches attempted to identify signaling processes in non-mammalian cells (primarily budding yeast).

ii. We take a more rigorous approach to evaluation than previous approaches. Since we focus on reconstructing a specific pathway, we can evaluate the precision and recall of our results by comparing them to a gold standard version of the pathway.

iii. We demonstrate that our approach can bridge the differences between database representations of a pathway. Indeed, we recover signaling interactions from two different databases that show surprisingly low overlap in the sets of proteins and interactions reported for a specific pathway.

iv. We discuss a novel approach for predicting pathway *crosstalk*, whereby two different pathways share common signaling proteins and interactions.

Our approach is reminiscent of related methods that attempt to connect a set of sources to a set of targets by using the edges in an interactome. Yeang *et al.* developed physical network models [139] to compute subnetworks that connect source proteins from a signaling pathway to downstream target genes. They annotated the interactome with variable attributes (e.g., sign of an edge to denote repression or activation) and predict their values. Physical network models identified novel signaling mechanisms in yeast [140] and proposed novel hypotheses for yeast signaling in response to DNA damage [137]. Subsequent papers

integrated expression data (with and without gene knockouts) with the underlying interactome and used integer [94] and linear programming [119, 141] to tackle the problem of reconstructing signaling pathways. Yosef *et al.* introduced ANAT, which balances both global (PCST) and local (shortest paths) formulations of the problem [143, 144]. Shih and Parthasarathy [113] proposed a heuristic to compute k partially edge-disjoint paths from causal proteins to target genes in yeast. We compare our method to a number of these techniques [119, 124, 141, 144].

6.2 Methods

Given only the receptors and transcriptional regulators in a specific signaling pathway, we hope to automatically reconstruct the precise collection of interactions from the pathway. Accordingly, we seek to compute a subset of the interactome that connects a set of receptor proteins S to a set of downstream transcriptional regulators T. In Section 6.2.1, we present an algorithm we call Linker that combines edge flux computations based on a teleporting random walk with a k-shortest loopless paths algorithm to identify relevant connections. We have successfully applied Linker to the problem of automatically expanding mathematical models of the cell cycle in budding yeast [97]. In this work, we demonstrate Linker's utility in a new application of reconstructing human signaling pathways. In Sections 6.2.2, 6.2.3, 6.2.4, and 6.2.5, we discuss previously-published approaches for discovering signaling pathways that we evaluate against Linker [119, 124, 141, 144]. Although not explicitly developed to recover signaling pathways, these approaches seek to connect a subset of nodes within a network, and we discuss how these methods can be applied for our purpose.

6.2.1 Linker

Linker is a two stage algorithm whereby we first compute edge traversal probabilities during a teleporting random walk followed by k-shortest paths that enumerate the most probable connections from a receptor to a downstream TR. Figure 6.1(a) illustrates this process (green arrows). Given a directed graph $G = (V, E)$ and set of receptor nodes S, we compute per-node visitation probabilities arising from a teleporting random walk commonly recog-

nized in the literature as PageRank [95]. Let W be the weighted adjacency matrix for G, where w_{uv} denotes the weight of the directed edge (u, v). We normalize edge weights such that $\sum_{x \in N_u^{\text{out}}} w_{ux} = 1$, where N_u^{out} is the set of out-neighbors of node u; thus, W is row stochastic. Consider the following random walk process. Place a random walker on any node in the graph, and allow the walker to transition from its current node u as follows:

Teleport: With probability $q/|S|$, she teleports to any receptor protein $x \in S$; the total probability of teleporting from u is q.

Walk: She walks from u to one of u's out-neighbors $v \in N_u^{\text{out}}$ with probability proportional to $(1 - q)w_{uv}$; thus, the total probability of walking to some out-neighbor of u is $1 - q$.

The parameter $q \in (0, 1]$ controls how often the walker teleports back to one of the receptors in S, thereby biasing the walker to paths that start at a node in S. As the number of steps in the random walk approaches infinity, the probability that the random walker visits each node converges to a stationary probability distribution if the graph is irreducible (i.e., every node is reachable from every other node) and aperiodic (i.e., the greatest common divisor of the lengths of the cycles in the graph is 1). In practice, the interactome we used contained dangling nodes (nodes with no outgoing edges). We add a directed edge from each dangling node to every other node in the graph; thus, the random walker has a small probability of moving from a dangling node to any node in the network during the next step of the random walk. After this modification, we verified that our network was strongly connected (ensuring the graph was irreducible) and contained cycles of length two and three (ensuring the graph was aperiodic). Let $\mathbf{p}$ be a column vector of visitation probabilities p_v for each node $v \in V$, and evaluate $\mathbf{p}$ as follows:

$$\mathbf{p} = \left(\mathbf{I} - (1 - q)W^{\mathrm{T}}\right)^{-1} q\mathbf{s}$$

where $\mathbf{I}$ is the $|V| \times |V|$ identity matrix and $\mathbf{s}$ is a column vector of teleportation probabilities. We set $s_v = 1/|S|$ for each node $v \in S$ and $s_v = 0$ otherwise. In practice, rather than inverting this large matrix, we use the well-known iterative power method to compute $\mathbf{p}$ efficiently. See Section 4.3.1 for a proof of convergence. We then assign a *flux score* f_{uv} to each edge as

follows:

$$f_{uv} = \frac{w_{uv} p_u}{d_u}$$

Thus, we distribute the node visitation probability p_u across the edges leaving u with probability proportional to each edge's weight. Edges with a high probability of being traversed receive a large edge flux while edges with a low probability of traversal receive a low edge flux. We call this entire procedure EdgeFlux.

KSP. Using the edge flux scores, we compute the k most probable loopless paths that begin at any receptor in S and terminate at any transcriptional regulator in T. We define the probability of a path to be the product of the edge flux scores along the path. Given a user-defined parameter k, we identify the k most probable paths as follows [142]. We add an artificial source s with a directed edge (s, x) for each node $x \in S$ and an artificial sink t with a directed edge (y, t) for each node $y \in T$. We assign the following cost to each edge (u, v):

$$c_{uv} = \begin{cases} -\log(f_{uv}) & \text{if } v \in V \setminus \{s, t\} \\ 1 & \text{if } u = s \text{ or } v = t. \end{cases}$$

Let the cost of a path be the sum of the costs of the edges in the path. Therefore, the least costly $s \rightsquigarrow t$ path is equivalent to the path from S to T that maximizes the product of edge flux scores along the path. In this modified graph, we identify the k shortest loopless paths from s to t using Yen's algorithm [142], which runs in $O\left(k|V||E| + k|V|^2 \log(|V|)\right)$ time.

6.2.2 ResponseNet

ResponseNet [141] is a flow-based algorithm to connect source and target proteins in an interaction network. Similar to other approaches, ResponseNet adds an artificial source node s and adds an edge (s, v) for every $v \in S$ and adds an artificial sink t and an edge (v, t) for every $v \in T$. Each edge (u, v) has a capacity c_{uv} representing the maximum amount of flow allowed to traverse that edge; we set all capacities to 1. After defining a variable f_{uv} representing the flow for edge (u, v), ResponseNet solves the following optimization problem:

$$\text{Objective: } \min_{f} \left(\left(\sum_{(u,v)\in E} -\log(w_{uv}) f_{uv} \right) - \left(\gamma \sum_{u \in S} f_{su} \right) \right)$$

s.t.

$$0 \leq f_{uv} \leq c_{uv} \quad \forall (u,v) \in E$$

$$\sum_{v \in V} f_{uv} = \sum_{v \in V} f_{vu} \quad \forall u \in V \setminus \{s,t\}$$

$$\sum_{v \in S} f_{sv} = \sum_{v \in T} f_{vt}$$

These constraints ensure that the flow does not exceed the capacity of each edge, the sum of incoming and outgoing flows are conserved across a node, and the amount of input flow to the network equals the amount of output flow. We implemented ResponseNet in Python, using CPLEX [27] to solve the linear program.

6.2.3 eQED

eQED is a linear program that models networks as electric circuits [119]. This method was originally developed to analyze expression quantitative trait loci (eQTL) data, for which eQED predicted the causal gene involved in a change in a gene's expression. eQED computes a score for each edge in the network, and we use these scores to propose candidate edges in a signaling pathway. The original eQED formulation only includes a single target node. We modified the algorithm to allow for multiple targets; we converted G into an electrical circuit as follows:

1. Assign each edge $(u,v) \in E$ a resistance $r_{uv} = 1$.
2. Assign each target $v \in T$ a voltage $\omega_v = 0$.
3. Add an artificial source s, and for each $v \in S$ add an undirected edge (s,v) with infinite resistance.

We introduced a source of current c into the network at the source node s. Knowing the resistances r_{uv} for all edges, we sought the voltages ω_v for each node $v \in V$ and the currents

c_{uv} for all edges $(u, v) \in E$. When G contained only undirected edges, the voltages and currents in the circuit produced by G could be solved using Kirchhoff's and Ohm's Laws. However, when G contained directed edges, eQED solved a linear program to preserve edge directionality. Let $D \subseteq E$ denote the directed edges in the network (i.e., $D = \{(u, v) | (u, v) \in E, (v, u) \notin E\}$). Define a variable d_{uv} for each directed edge $(u, v) \in D$. eQED solved the following linear program:

$$\text{Objective: } \min \sum_{(u,v)\in D} (d_{uv} - (\omega_u - \omega_v))$$

$$\text{s.t.}$$

$$\sum_{(s,v)\in E} c_{sv} = c$$

$$\forall\, u, v \in T : \omega_u = \omega_v$$

$$\forall\, (u, v) \notin D : c_{uv} = \frac{(\omega_u - \omega_v)}{r_{uv}}$$

$$\forall\, (u, v) \in D : \begin{cases} c_{uv} = \frac{d_{uv}}{r_{uv}} \\ d_{uv} \geq \omega_u - \omega_v \\ d_{uv} \geq 0 \end{cases}$$

$$\forall\, v \in V \setminus \{t\} : \sum_u c_{uv} = 0$$

Intuitively, d_{uv} is a variable that either equals 0 or $\omega_u - \omega_v$, depending on the directionality of the edge. eQED returned a solution that attempted to preserve Ohm's law by enforcing a penalty if the change in voltage between u and v was different from d_{uv}. eQED computed a current for each edge. We implemented eQED in Python and used CPLEX [27] to solve the linear program.

6.2.4 PCSF

Recently, Tuncbag *et al.* presented a method to recover signaling pathways using a Prize-Collecting Steiner Forest (PCSF) approach [124]. Given a graph G with prizes b_v on the nodes and costs c_e on the edges, PCSF finds a forest $F = (V_F, E_F)$, where $V_F \subseteq V$

and $E_F \subseteq E$, that minimizes the following objective function:

$$\Phi(F) = \sum_{v \notin V_F} b_v + \sum_{e \in E_F} c_e + \omega k$$

where ω was a parameter that we set to 1 in this work, and k was the number of trees in the forest F. The first term in the function enforced a penalty for each prize that was not collected because the node was not included in F. The second term penalized Φ by the total cost of the edges in the solution. Finally, the third term ensured that the number of trees in F remained small. To solve PCSF, the authors transformed the original graph G into an instance of the Prize-Collecting Steiner Tree (PCST) problem and applied PCST solving software, MsgSteiner, to approximate a minimal solution to PCST as follows [7]. Add an artificial source s with directed edges (s, v) for each $v \in S$, and set the cost of each added edge to ω. Assign a positive prize $b_v = \rho$ to every target node $v \in A$ and set the prize of all other nodes to 0.

MsgSteiner produces a tree rooted at s. Removing s from this tree yields a forest of trees, with each tree rooted at one of the source nodes. This forest includes a target node if the prize ρ is large enough to compensate for the cost of the path to reach the target. We ran PCSF with different values for ρ ranging from 1 to 10. MsgSteiner takes a number of parameters. We set the tree depth to 10, the reinforcement parameter to 10^{-3}, the penalty for the number of trees (ω) to 1, and random noise to 10^{-6}. Please refer to the MsgSteiner documentation for details regarding these parameters [7]. Note that MsgSteiner discovers Steiner trees with directed edges pointing *toward* the root of the tree. Therefore, we reversed the edge directions in the original graph G before running MsgSteiner and flipped them back before reporting the forest.

6.2.5 ANAT

ANAT is a recently-published approach designed to identify connections between a given source node s and a set of target terminal nodes T [144]. The method balances the trade-offs between local and global formulations of the problem to identify a connected subnetwork $H = (V_H, E_H)$, where $V_H \subseteq V$ and $E_H \subseteq E$, such that $s \in V_H$ and $T \subseteq V_H$. Letting $P(s, t)$

denote the set of edges in the path from s to t in H, the local formulation seeks a subgraph H that minimizes the following:

$$F_L(H) = \sum_{t \in T} \sum_{e \in P(s,t)} \omega_e$$

Thus, $P(s,t)$ is simply the shortest path from s to t in G, and H is the union of the shortest paths from s to each target in T. The global formulation seeks a subgraph H that minimizes the following:

$$F_\Gamma(H) = \sum_{e \in E_H} \omega_e$$

Minimizing this formulation is equivalent to finding a minimum Steiner tree that connects the set of nodes $T \cup \{s\}$. ANAT balances these two formulations by minimizing their linear combination $F = cF_\Gamma + F_L$. They set c to $\hat{F}_L / \hat{F}_\Gamma$, where $\hat{F}_L$ and $\hat{F}_\Gamma$ are the optimal values of F_L and F_Γ, respectively.

In practice, the authors minimized F using CHARIKAR-α, an extension to an approximation algorithm for the minimum Steiner tree problem [20]. CHARIKAR-α controls emphasis on the local versus global formulation through a parameter $\alpha \in [0, 0.5]$. Values of α near 0 provide approximations close to the local formulation, while values of α near 0.5 provide approximations close to the global formulation. In our comparisons to ANAT, we used $\alpha \in \{0, 0.25, 0.5\}$.

6.3 Datasets

Human interactome. We constructed a directed human molecular interaction network that included undirected physical interactions from the following databases: BIND, DIP, InnateDB, IntAct, MINT, MatrixDB, and Reactome. We also included directed phosphorylation and transcriptional regulatory interactions from NetPath [57], SPIKE [96], and a study by Vinayagam *et al.* [133]. We converted all undirected interactions to bi-directional edges and removed self loops. The network contained 11,266 nodes, 56,554 bi-directed edges, and 16,792 directed edges (129,900 total directed edges).

NetPath pathways. We collected interactions reported in the NetPath human signaling pathways [57]. At the time of writing, NetPath contained 28 signaling pathways listed in Table 6.1. To identify the set of signaling receptors in each pathway, we computed the intersection of the proteins in the pathway with a previously-published list of human signal receptors [3], and we manually inspected these lists for erroneous or missing receptors for each pathway. Since we sought signaling mechanisms that cascade to downstream transcriptional events, we defined the targets of each pathway as the set of transcriptional regulators (TRs) that participate in the pathway. We retrieved the set of human TRs reported in two studies: i) all TRs listed by Ravasi *et al.* [102] and ii) high-quality TRs from Vaquerizas *et al.* [131] which they classify as "a", "b", and "other" (omitting classes "c" and "x"). TRs classified as "a" and "b" had experimental evidence of regulatory function in a mammalian organism, and 27 manually-curated TRs from other sources were classified as "other".

We utilized the largest weakly connected component of each NetPath pathway. Of the 28 pathways, we retained 16 pathways for analysis that met the following criteria: i) the largest weakly connected component of the pathway contained at least one receptor and one TR, ii) the sets of receptors and TRs were disjoint, and iii) the minimum cut between the receptors and TRs was at least three (i.e., three edges must be removed from the pathway to disconnect the receptors from the TRs). The first two criteria ensured that the pathways had a natural beginning and end to the signal propagation. The Notch pathway was the only one that did not meet these criteria. The third criteria ensured the pathway was sufficiently connected. We included the third criteria because several excluded pathways had a minimum cut of zero; such curated pathways were likely highly incomplete as there was no connection (path) from any signaling receptor to a downstream TR. Table 6.1 highlights pathways in blue that did not meet third criteria of connectivity.

KEGG pathways. We collected interactions reported in the Kyoto Encyclopedia of Genes and Genomes (KEGG) database [60] for five pathways that also appeared in our list of analyzed NetPath pathways: BCR, EGFR1, Hedgehog, TGFβ Receptor, Wnt. We preprocessed the KEGG pathways using the same procedure that we applied to the NetPath pathways.

Pathway	Pathway Statistics					
	Edges	Nodes	Rec	TRs	SP	MC
$\alpha 6\beta 4$ Integrin	227	66	2	2	∞	0
BCR	475	137	2	17	2.00	8
BDNF	146	73	3	4	1.00	4
CRH	64	24	2	8	∞	0
EGFR1	1491	231	3	32	1.00	30
FSH	32	20	1	1	∞	0
Hedgehog	138	36	2	12	∞	0
IL1	184	43	1	5	3.00	7
IL2	256	67	3	11	1.00	16
IL3	187	70	2	9	2.00	5
IL4	187	58	3	11	1.00	2
IL5	76	32	2	3	1.00	2
IL6	177	53	2	13	1.00	6
IL7	57	18	2	3	1.00	5
IL9	34	14	2	3	2.00	1
Kit Receptor	216	75	1	8	1.00	5
Leptin	150	55	1	14	1.00	8
Notch	281	74	4	22	1.00	25
Oncostatin M	97	38	3	12	1.00	2
Prolactin	210	70	1	10	1.00	10
RANKL	154	58	1	11	1.00	4
TCR	525	154	2	19	1.00	5
TGFβ Receptor	940	208	2	75	1.00	32
TNFα	959	239	2	44	1.00	33
TSH	96	50	1	5	1.00	2
TSLP	18	7	2	2	1.00	2
TWEAK	35	17	1	4	∞	0
Wnt	453	107	12	13	2.00	7

Table 6.1: NetPath pathway statistics. Abbreviations: receptors (Rec), transcriptional regulators (TRs), shortest path from any receptor to any TR (SP), minimum receptor-TR cut (MC). MC is the minimum number of edges that must be removed from the pathway to eliminate all paths from a receptor to a TR. Bold pathways were used for our analysis. We ignored pathways with MC less than 3 (blue) and pathways whose receptors also appeared as downstream targets (green).

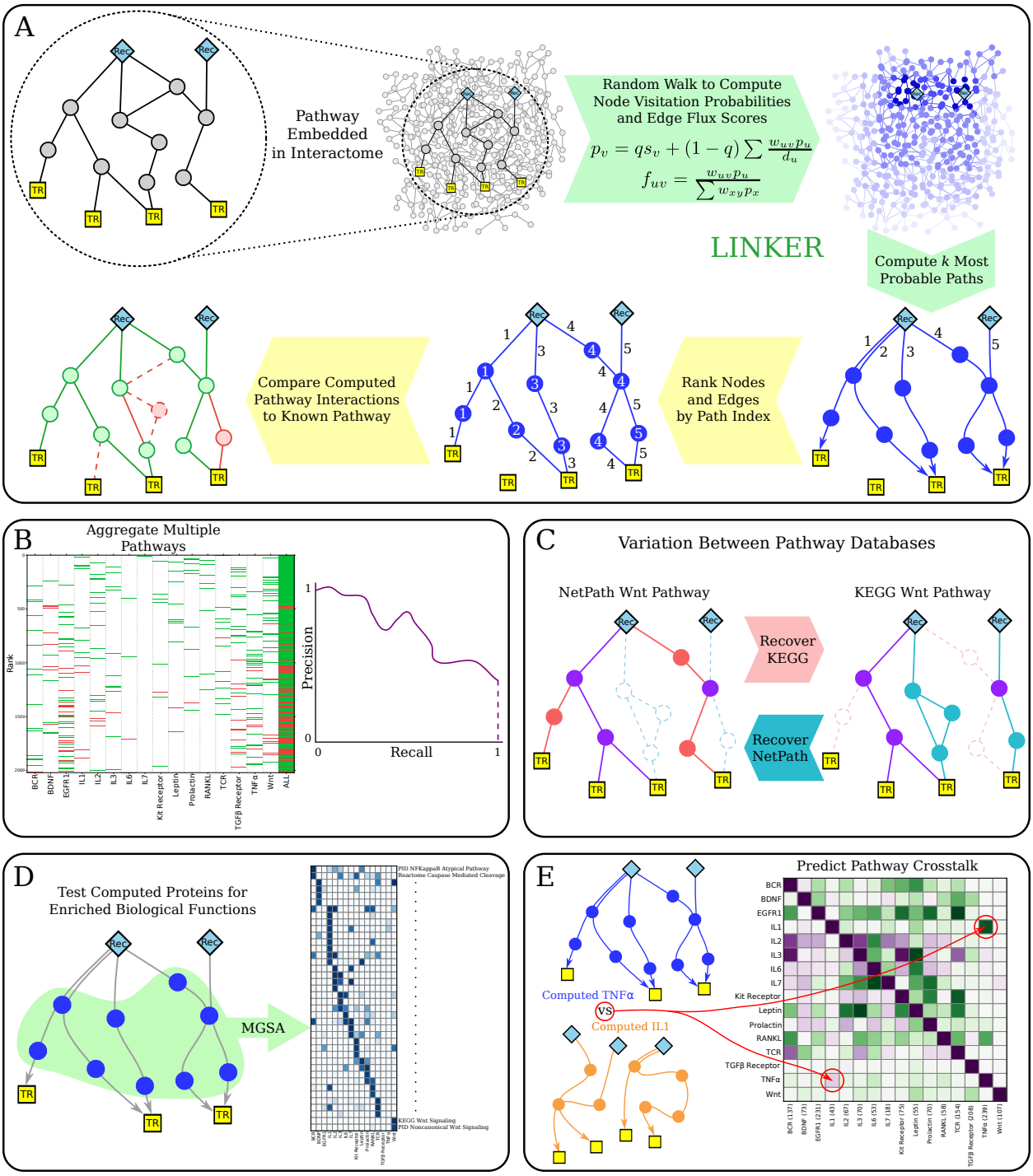


Figure 6.1: (*a*) Overview of the LINKER approach for reconstructing signaling pathways. (*b*) Precision-recall analysis compares computed connections to NetPath pathway representations. (*c*) Test how well we can recover KEGG pathway representations from NetPath and *vice versa*. (*d*) Internal proteins in the computed pathways are significantly enriched in relevant biological functions from MSigDB. (*e*) Compare the paths computed by LINKER for two different pathways; pairs of pathways that utilize similar protein sets may indicate crosstalk.

6.4 Results

We tested the ability of each algorithm to reconstruct the 16 NetPath pathways when given a background network containing the pathway as a subnetwork and the sets of receptors and transcriptional regulators (TRs) for each pathway. Figure 6.1(a) provides an overview of Linker and Figures 6.1(b–e) summarize our primary analyses and results. In Section 6.4.1, we evaluate our computed connections against the NetPath interactions for each pathway, and we compare our approach to a variety of popular methods in the literature. Since different pathway databases often represent the same pathway using considerably different collections of interactions, Section 6.4.2 addresses our ability to recover pathway representations between databases. In Section 6.4.3, we evaluate the proteins from our computed connections for significantly enriched biological functions. Lastly, Section 6.4.4 discusses our approach for predicting crosstalk between pathways.

6.4.1 Reconstructing NetPath Pathways

A prevailing trend in the literature represents signaling pathways as the set of proteins involved in the pathway [63, 73]. We independently assessed recovery of pathway proteins (nodes) and interactions (edges) by each approach. Here, we demonstrate the difference in the difficulty of recovering the precise collection of interactions from a pathway versus recovering only participating proteins.

We used Linker to compute the 5000 most probable paths from the receptors to the TRs of each pathway. We compared our proposed interactions to the following algorithms in the literature: eQED [119], ResponseNet [141], PCSF [124], and ANAT [144]. We also included EdgeFlux and KSP, the individual components of Linker, as separate algorithms. Lastly, we compared our results to Degree, a naïve method that ranks nodes and edges by their degree and uses these rankings to make predictions for each pathway. A recent series of publications by Gillis and Pavlidis demonstrated that a simple ranking of nodes by their degree in a protein interaction network can predict gene function with results comparable to more sophisticated network-based approaches [38, 39]. We included Degree to assess whether this method could effectively reconstruct interactions within

specific signaling pathways. Table 6.2 summarizes the values used to rank proposed proteins and interactions from each algorithm.

Algorithm	**Nodes**	**Edges**
Linker and KSP	Index of the first path using each node	Index of the first path using each edge
EdgeFlux	Node visitation probabilities	Edge flux scores
eQED	Sum of incoming current	Absolute value of edge current
ResponseNet	Maximum flow of adjacent edges	Flow along the edge
ANAT and PCSF	Inclusion in computed subnetwork	Inclusion in computed subnetwork
Degree	Degree in the interactome	Sum of the degrees of incident nodes

Table 6.2: Summary of values used to rank nodes and edges by each algorithm.

Recovering signaling proteins. We ranked predicted proteins and interactions from the interactome according to each algorithm and computed precision-recall curves. As we walked down a ranked list of predictions for a pathway, *recall* denotes the fraction of positives recovered while *precision* denotes the fraction of correct predictions at each rank. For each pathway, we defined positive proteins as the set of proteins in the NetPath pathway. We defined the sets of negative proteins in the following two ways:

i. "ignore none" – all proteins from the interactome that are not in the NetPath pathway

ii. "ignore adjacent" – all proteins v from the interactome such that v does not interact with any positive protein from the NetPath pathway

In the second case, we ignored proteins that were adjacent to the pathway. We randomly selected negatives to ensure a $1 : 50$ ratio of positives to negatives in both cases. We constructed the positive and negative sets for each pathway independently, and we merged the computed interactions across all pathways based on the prediction score reported by each algorithm (see Table 6.2). Note that each protein from the interactome may appear up to 16 times in this aggregated ranked list, since a protein received a different prediction score for each pathway. We only considered a protein from a NetPath pathway as a positive protein during the recovery of that pathway. Thus, many proteins were treated as a positive once and as a negative for the recovery of other pathways.

Figure 6.2 illustrates the precision-recall curves for each algorithm aggregated across the 16 analyzed NetPath pathways. Dashed vertical lines indicate the last value of recall

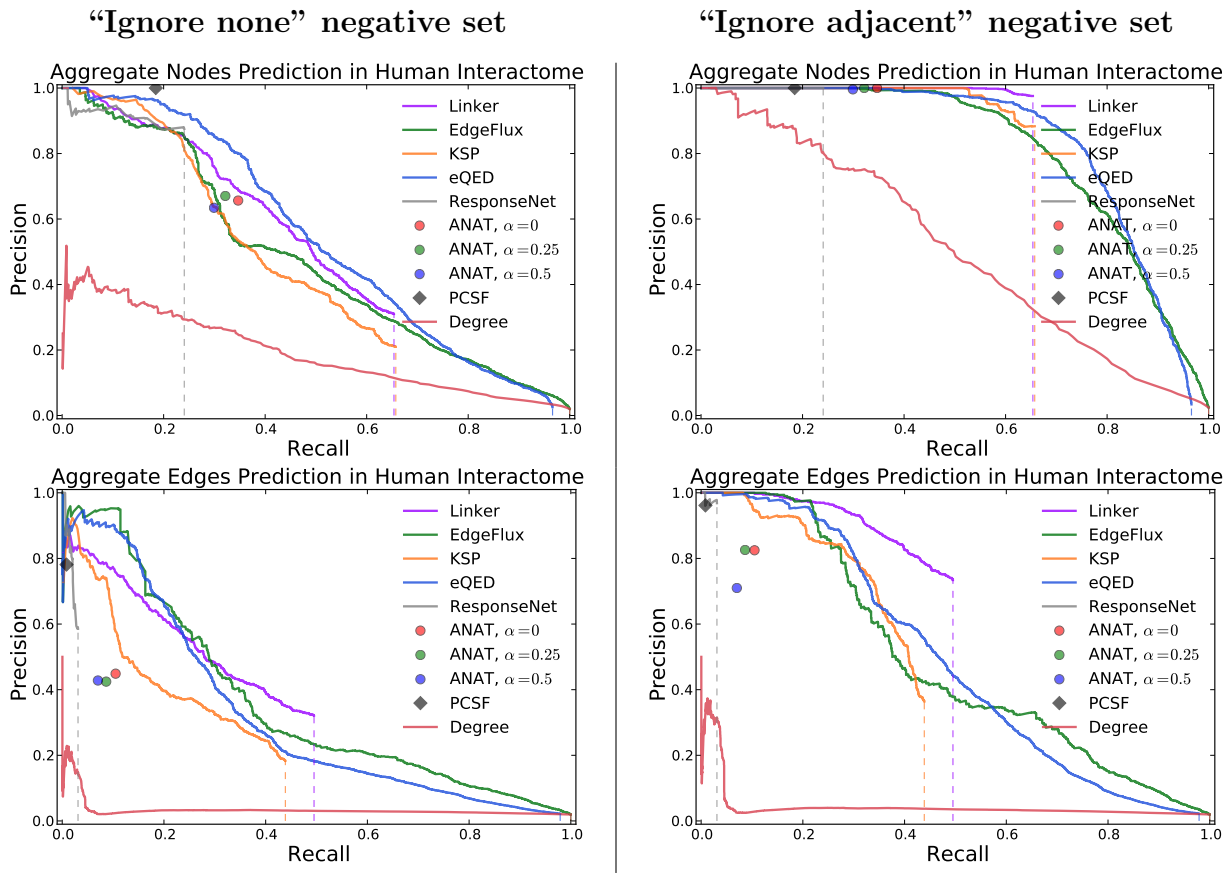


Figure 6.2: Aggregated precision-recall curves for reconstructing proteins (top row) and interactions (bottom row) from NetPath pathways. In the left column, the set of negative interactions includes all nodes (top) or edges (bottom) that do not participate in the pathway; in the right column, we exclude nodes (top) or edges (bottom) that are adjacent to the NetPath pathway.

reached by an algorithm, as some algorithms do not rank all edges in the network. For example, we computed 5000 paths using LINKER, but there is no guarantee that all positive nodes or edges were used in the top 5000 paths. We also note that since ANAT and PCSF compute a subnetwork rather than rank nodes and edges, we could only compute a single value of precision and recall for those methods. The top row of plots in Figure 6.2 demonstrates the recovery of proteins from the signaling pathways. The left and right plots use the "ignore none" and "ignore adjacent" definitions of negative proteins, respectively. First, note that DEGREE was a poor predictor of proteins that participated in each signaling pathway. eQED outperformed LINKER slightly when we did not ignore any proteins from the set of negatives. However, after ignoring adjacent proteins, LINKER achieved nearly

perfect precision up to ∼65% recall, slightly outperforming eQED. Thus, many proposed proteins deemed false positives in the "ignore none" precision-recall curve directly interact with proteins from the corresponding NetPath pathway. Note that EdgeFlux or KSP alone performed worse than Linker in both cases. Nonetheless, several algorithms achieved nearly perfect recovery of proteins participating in the signaling pathway when we excluded adjacent proteins from the collections of negatives.

Recovering signaling interactions. We contrast the protein-based analysis with the problem of reconstructing the precise interactions that participate in each NetPath pathway. The bottom row of plots in Figure 6.2 are precision-recall curves for the recovery of signaling interactions (analogous to the top two plots). We defined the set of positive interactions as the set of unordered interacting pairs of proteins from the NetPath pathway. For this analysis we ignored the issue of correctly predicting directionality and evaluated how well each algorithm recovers the sets of interacting proteins. Similar to our analysis of protein recovery, we defined the sets of negative interactions for each pathway in two ways:

i. "ignore none" – all interacting pairs of proteins from the interactome that do not interact in the NetPath pathway

ii. "ignore adjacent" – all interacting pairs (u, v) such that both u and v are not in the NetPath pathway

Similar to the definition of negative nodes, the second case ignores interactions adjacent to the pathway.

The lower precision in the bottom two plots compared to the top two plots of Figure 6.2 demonstrates that recovering interactions is a much more difficult problem than solely identifying participating signaling proteins. When ignoring no interactions in the set of negatives, EdgeFlux and eQED outperformed the other approaches up to ∼25% recall, at which point Linker achieved similar precision scores. Between 35% recall and 50% recall, Linker dominated the other approaches. After ignoring adjacent interactions, Linker, EdgeFlux, and eQED achieved greater than 95% precision up to ∼20% recall, at which point Linker outperformed all other approaches up to 50% recall. Note that Linker achieved ∼75%

precision at 50% recall, much greater than that of eQED (45% precision), the second place algorithm.

The drastic improvement in precision after ignoring proteins or interactions adjacent to each pathway indicated that many of the proposed interactions that were flagged as false positives were incident on the NetPath pathway. Since these manually-curated pathways are highly incomplete, many of these interactions may indeed represent valid pathway interactions that have not yet been added to the pathway through the curation process. High-confidence predictions adjacent to the pathway are ideal candidates for further experimental study in expanding known signaling pathways.

Stratifying aggregate predictions. The precision-recall analysis from the previous section aggregated predictions across all pathways. To further investigate how each pathway contributed to the aggregate curves, we developed a visualization of the predictions used to construct the precision-recall curves at a specific value of recall. The precision-recall maps in Figure 6.3 display the aggregate LINKER interaction rankings up to recalls of 10% (top) and 50% (bottom) for the "ignore adjacent" plot (from bottom right of Figure 6.2). The far right column of each map visualizes the aggregated rankings. The most confident predictions are at the top of the map, with the least confident predictions on the bottom. Green and red horizontal bars indicate true positive and false positive predictions, respectively.

The first 16 columns of each map stratify the predictions from the individual pathways, indicating for which pathway each prediction was made. By inspecting this stratification, we can quickly identify the pathways contributing to the false positive predictions in the aggregate precision-recall curve. The top map in Figure 6.3 demonstrates that that up to 10% recall, no false positive predictions were made by LINKER. The bottom map in Figure 6.3 illustrates the aggregate false positive and negative predictions up to a recall of 50%. The stratification of these predictions indicates that the EGFR1 and TGFβ Receptor pathways (and to a lesser extent TNFα and Wnt) were the primary culprits that introduced false positive predictions. Further analyzing these maps may facilitate discovery of classes of pathways that are inherently difficult to predict, perhaps due to certain topological properties or incomplete manual curation.

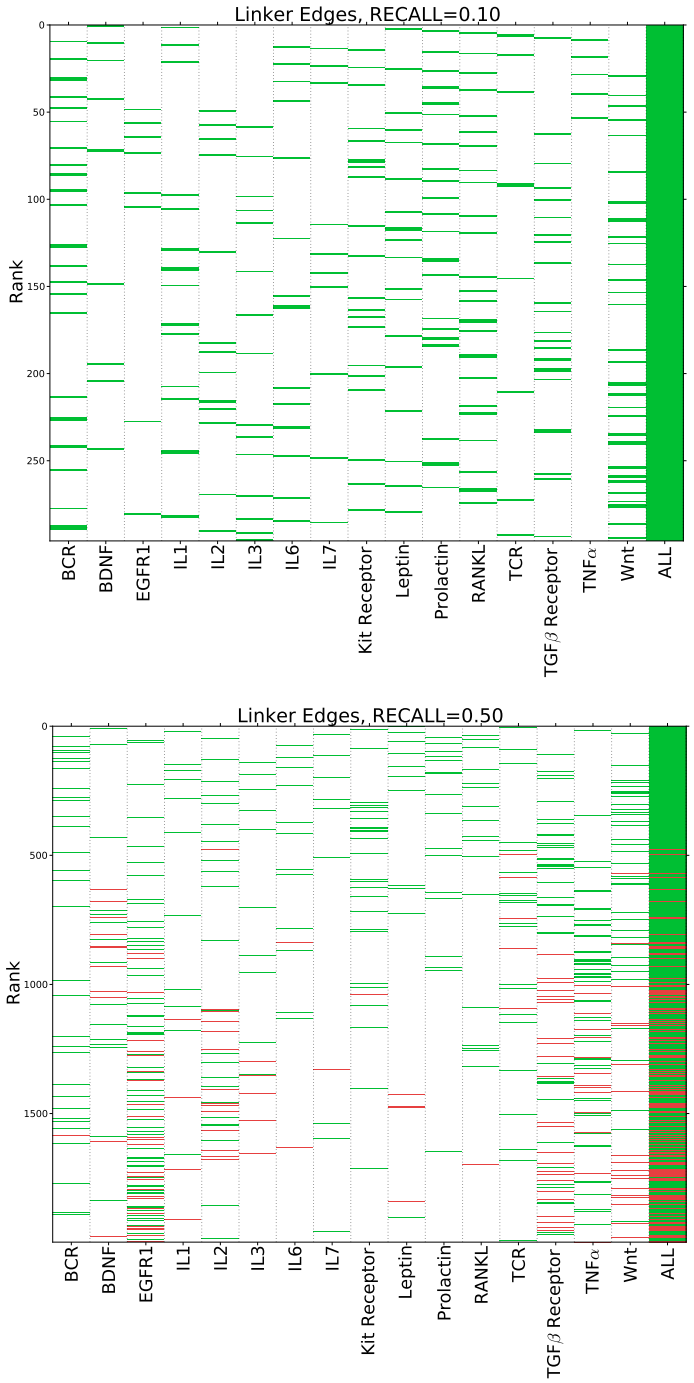


Figure 6.3: Maps of aggregated predictions by LINKER across all NetPath pathways. Predictions are shown for 10% recall (top) and 50% recall (bottom), ranked from top to bottom of each map. Green and red ticks indicate true positive and false positive predictions, respectively. Aggregated predictions from the far right column are broken down into the respective pathways across the first 16 columns.

6.4.2 Reconstructing Pathways From Other Databases

Representations of signaling pathways in databases such as NetPath are the result of a time-consuming manual curation process. Depending on the amount of curation and availability of details about the pathway from the literature, some pathways may be highly incomplete. Though we used NetPath interactions as positives in the previous section, we note that many interactions relevant to each pathway may be missing from NetPath. Indeed, alternative manually-curated signaling pathway databases vary greatly in the sets of proteins and interactions included for each pathway. For example, the NetPath Wnt pathway contains 107 proteins and 428 interactions, while the KEGG representation of the same pathway contains 128 proteins and 1042 interactions. Moreover, the two databases only share 57 proteins (Jaccard index of 0.32) and 98 interactions (Jaccard index of 0.07) between their representations. We anticipated that many interactions proposed during our reconstruction of NetPath pathways may have been considered negatives during the evaluations from the previous section even though they are supported by KEGG.

We assessed how well Linker recovered the interactions from each KEGG pathway using the receptors and TRs from the NetPath representation of the same pathway (and vice versa). To this end, we collected the interactions from five pathways available in both KEGG and NetPath databases (see Section 6.3). Similar to our procedure for reconstructing pathways for NetPath pathways, we used Linker to compute 5000 paths connecting the receptors to the TRs reported by KEGG for each of the five pathways. We evaluated the proposed connections using the same precision-recall analysis from the previous section. For each pathway, we separately considered interactions reported by NetPath, KEGG, and their union as the set of positives for computing precision and recall; thus, we constructed three precision-recall curves for each set of proposed interactions. As before, we used the "ignore adjacent" randomly-selected pairs of interacting proteins as negatives.

Figure 6.4 contains the resulting precision-recall curves. The solid curves correspond to edge rankings given by Linker when the receptors and TRs from NetPath were provided as input; thus, the same ranked list of edges was used for the three solid curves. Similarly, the dashed curves correspond to edge rankings predicted by Linker using the receptors and

TRs reported by KEGG as inputs. Purple, green, and blue curves denote whether positives were taken from NetPath, KEGG, or their union, respectively. When NetPath receptors and TRs were provided to LINKER (solid curves) as input, we consistently achieved higher precision for predicting NetPath interactions over KEGG. Nonetheless, LINKER still managed to achieve reasonably high precision for predicting KEGG interactions (solid green curve) from the NetPath inputs; precision remained above 70% up to 25% recall. Moreover, when the union of NetPath and KEGG defined the set of positive interactions, we see equivalent or higher precision over either NetPath or KEGG alone. The increase in precision indicates that many of the interactions computed by LINKER that were deemed negatives by one database were supported by the other database. This result suggests that predicted interactions not supported by NetPath are ideal candidates for expanding the existing pathway, as many of these interactions are supported by KEGG.

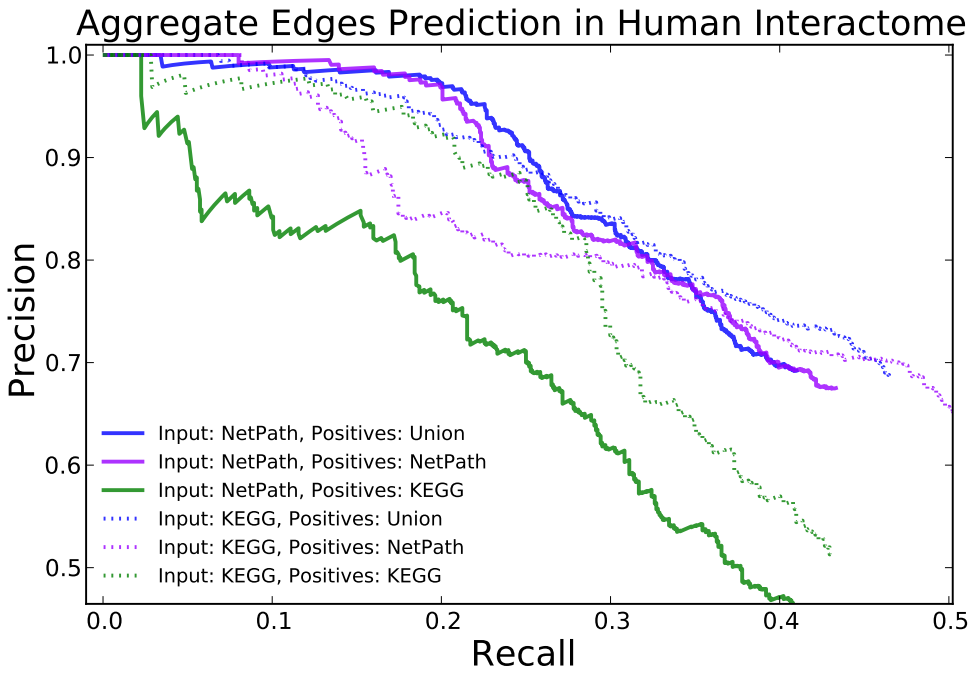


Figure 6.4: Aggregated precision-recall curves for recovering interactions for five pathways represented in the NetPath and KEGG databases. We provided receptors and TRs from NetPath (solid) and KEGG (dashed) to LINKER to reconstruct each pathway. We defined positives as interacting pairs of proteins from NetPath (purple), KEGG (green), or their union (blue). Thus, the same list of ranked interactions constructed the three dashed curves (and three solid curves); only the definition of positives changes. **Note:** Axes are truncated at 0.5.

Figure 6.5 visualizes the top 200 paths computed by LINKER connecting NetPath signal receptors to TRs. This network illustrates the vast difference between NetPath and KEGG pathway representations and our ability to recover interactions from both when prior knowledge from only one of the databases was provided to our algorithm. Nodes in this network indicate signal receptors (vees), downstream TRs (rectangles), and pathway internal nodes (ovals). Nodes and edges are colored pink, blue, or purple if they are supported by representations of the Wnt pathway in NetPath, KEGG, or both databases, respectively; green nodes and edges denote proteins and interactions not supported by either NetPath or KEGG. Notice that NetPath supported ten proteins (three of which are downstream TRs) included in the paths from LINKER were supported by NetPath but were missing from the KEGG database. Similarly, ten internal predicted proteins were included in KEGG but were missing from the NetPath representation of the Wnt pathway. The interactions proposed by LINKER told a similar story. Only a few interactions were supported by both databases, but nearly all predicted interactions participated in either the NetPath or KEGG representation.

As shown in Figure 6.5, the NetPath Wnt pathway includes R-spondin family proteins (RSPO1 and RSPO3) that were included in the top 200 paths computed by LINKER. However, these proteins did not belong to KEGG's representation of the Wnt pathway despite evidence that R-spondins are crucial regulators of the canonical Wnt signaling pathway [52, 64]. Similarly, the phospholipase C β proteins (PLCB1/2/3/4) were missing from the NetPath Wnt pathway but were included in KEGG. While the canonical Wnt signaling pathway is primarily driven by accumulation of β-catenin in the cytoplasm, the non-canonical Wnt-Ca^{2+} pathway represents one of the β-catenin-independent forms of Wnt signaling [91]. It is known that the phospholipase C proteins play an important role in the release of Ca^{2+}. Regulation of this non-canonical pathway has been linked to cancer, inflammatory response, and neurodegeneration [91]. NetPath curators may have purposely excluded these proteins and their incident interactions from the Wnt signaling pathway as NetPath focuses primarily on the β-catenin-dependent canonical pathway. Nonetheless, we demonstrated that many of the proteins and interactions included by LINKER are highly relevant to Wnt signaling despite not being included in NetPath's representation of the pathway.

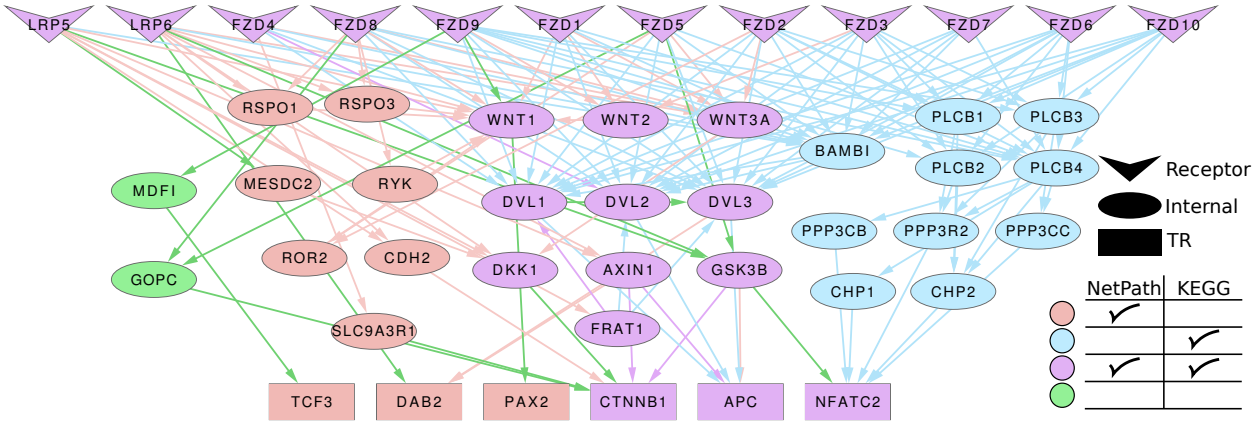


Figure 6.5: Top 200 paths predicted by LINKER that connect NetPath signaling receptors to downstream TRs. Nodes indicate signal receptors (vees), downstream TRs (rectangles), and pathway internal nodes (ovals). Pink, blue, and purple nodes and edges indicate proteins and interactions supported by representations of the Wnt pathway from NetPath, KEGG, or both databases, respectively. Green nodes and edges denote proteins and interactions not supported by either NetPath or KEGG.

6.4.3 Functional Enrichment of Pathway Predictions

We used functional enrichment tests to identify predefined biological functions (i.e., sets) that were significantly enriched in our reconstructed NetPath pathways. A variety of functional enrichment methods from the literature perform a term-by-term analysis, reporting the significance of the overlap between each function and the collection of proteins being studied [13, 65, 98, 118]. One disadvantage of these approaches is that they often return long lists of significantly enriched functions that must be further post-processed through manual inspection. Moreover, many of the top-ranking functions are difficult to distinguish, as the functions themselves may be highly similar and annotate nearly identical protein sets.

Model-based Gene Set Analysis (MGSA) [10] simultaneously analyzes the entire collection of functions and incorporates the similarity of annotations between functions into its analysis. MGSA seeks a set of non-overlapping functions that explain a given studyset (observed set of proteins). MGSA computes a posterior probability for each pathway that reflects how well the pathway overlaps with the study set while not overlapping with other pathways with higher posterior probability. This method accepts two parameters α and β that control the fractions of unknown false positive and false negative proteins, respectively. We set the upper limit of α to 0.3 and an upper limit of β to 0.5. Thus, MGSA seeks a set of functions

such that fewer than 30% of the proteins annotated by the selected functions are not in the studyset, while at least half of the studyset is annotated by at least one of the selected functions.

Functions enriched in predicted pathways. We applied MGSA to the top 250 proteins computed by Linker for each NetPath pathway. We ranked proteins by the index of the first path in which they appeared according to Linker. We tested the 250 proteins predicted from each pathway against 1452 gene sets from MSigDB [73]. We used the canonical pathways reported in the C2 collection of MSigDB 3.1, which only includes curated biological processes. The heatmap in Figure 6.6 visualizes functions that were enriched with an MGSA posterior probability greater than 0.95 for at least one NetPath pathway. Some functions were enriched in the predictions for multiple pathways, but the heatmap demonstrates the prevailing trend that most of the highly-enriched functions were specific to a particular NetPath pathway. Many of the predicted pathways are enriched in the same pathway representation in alternative pathway databases, strengthening our claim from the previous section that Linker can predict signaling proteins across different databases. Indeed, the top 250 predicted proteins for the Wnt pathway are highly enriched in the KEGG Wnt signaling pathway as well as the PID [90] canonical and noncanonical Wnt signaling pathways, each with a posterior probability greater than 0.98.

Novel enriched functions. We also applied MGSA to the top 250 nodes predicted for each pathway after excluding proteins that participate in the known NetPath pathway. We sought novel functions enriched in proteins related to each pathway that could not be discovered by performing functional enrichment analysis on the NetPath protein sets. Figure 6.7 illustrates the results of this modified enrichment test. Notice that after ignoring proteins internal to each NetPath pathway some of the more obvious functions are no longer highly enriched. For example, the KEGG and PID Wnt pathways were not highly enriched in the Wnt predictions. As expected, this suggested that the Wnt signaling paths predicted by Linker shared many proteins with representations of the Wnt pathway from other databases. Some functions were enriched even after ignoring the NetPath proteins. As

an example, we highlight "KEGG Adherens Junction" which was enriched in the NetPath Wnt pathway predictions before and after ignoring the internal proteins to each pathway.

KEGG Adherens Junction. The canonical Wnt signaling pathway includes the Frizzled (FZD) family of receptor proteins which receive the Wnt ligand as a signal. This binding initiates the pathway by activating the Dishevelled protein (DVL), inhibiting GSK3, and rendering GSK3 unable to mark β-catenin (CTNNB1) for degradation. Free β-catenin ultimately accumulates, enters the nucleus, and activates the TCF/LEF family of transcription factors. β-catenin maintains two primary functions within the cell: cell adhesion and transcriptional regulation, thereby supporting the known connection between Wnt signaling and cell adhesion [15, 17]. During cell adhesion, β-catenin binds to the cytoplasmic tails of cadherin molecules, which constitute the core components of adherens junctions [46]. Conversely during transcription, β-catenin translocates to the nucleus where it interacts with the LEF and TCF family of transcription factors [11, 93].

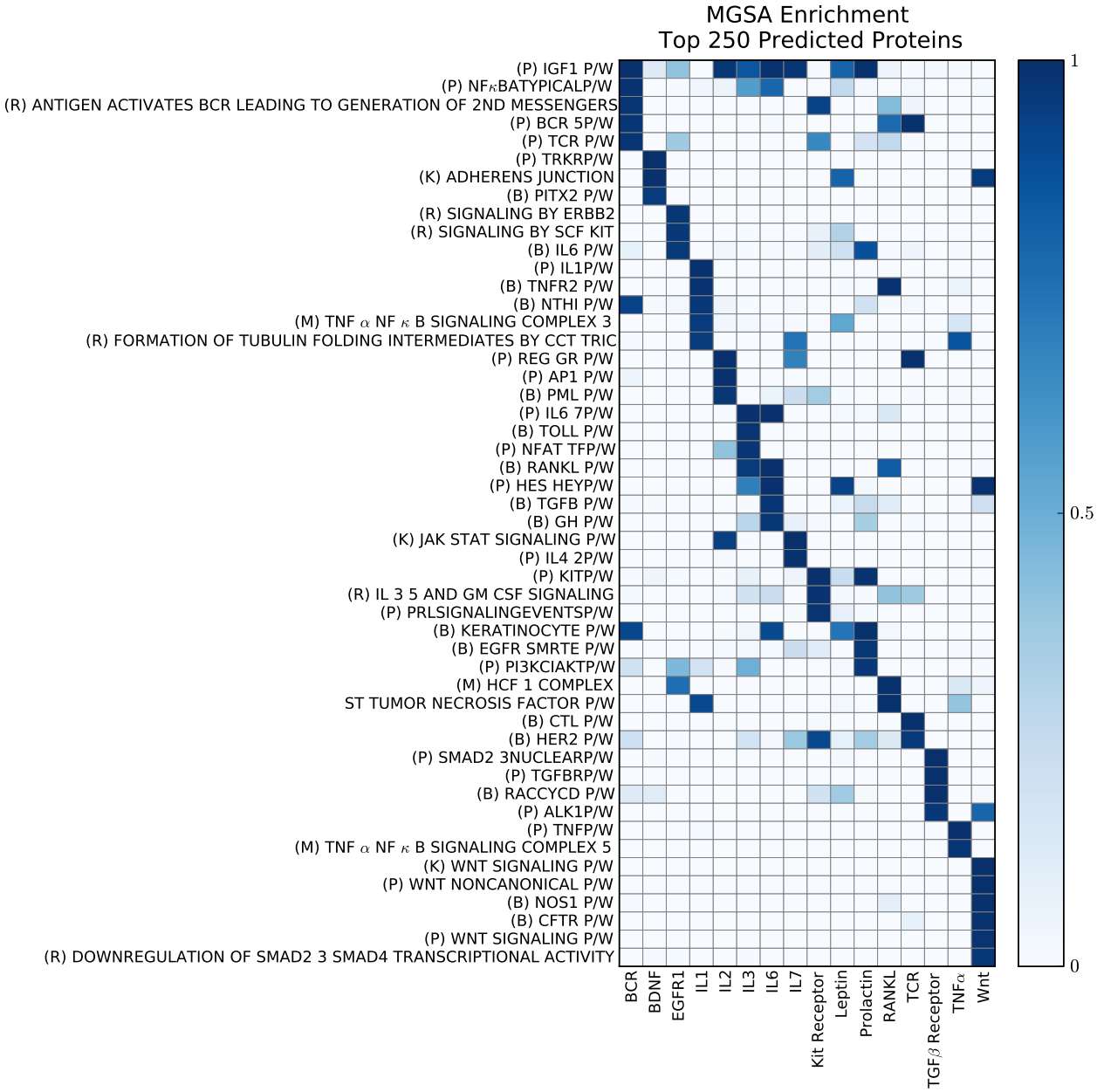


Figure 6.6: MGSA enrichment of MSigDB gene sets in the top 250 proteins predicted by Linker for each pathway. The heatmap includes gene sets with an MGSA posterior probability of at least 0.95 for any pathway. P/W stands for pathway; the source database for each function is shown in parentheses: KEGG (K), Reactome (R), MIPS (M), BioCarta (B), PID (P).

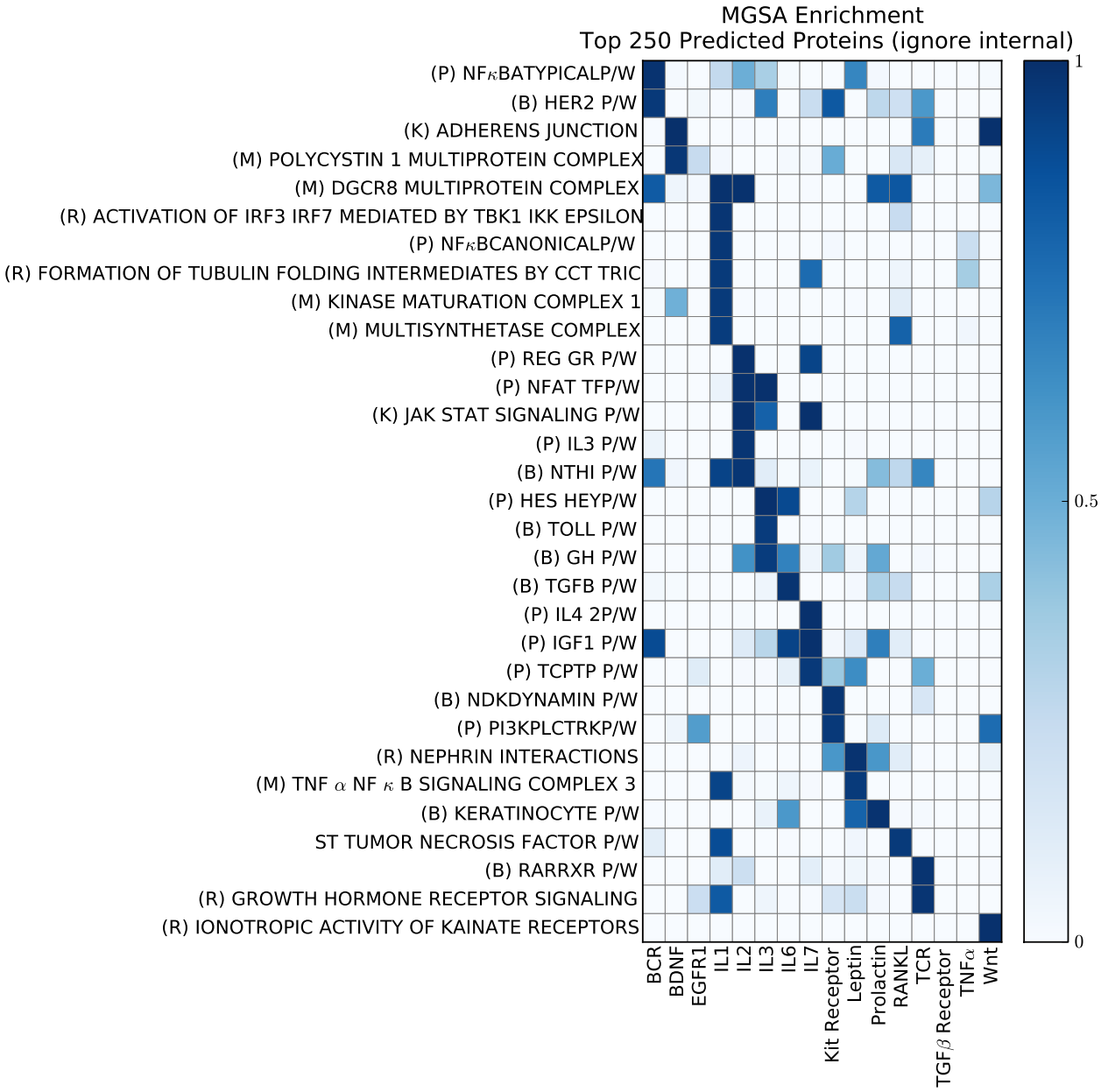


Figure 6.7: MGSA enrichment of MSigDB gene sets in the top 250 proteins predicted by Linker after ignoring proteins from the corresponding pathway. The heatmap includes gene sets with an MGSA posterior probability of at least 0.95 for any pathway. P/W stands for pathway; the source database for each function is shown in parentheses: KEGG (K), Reactome (R), MIPS (M), BioCarta (B), PID (P).

6.4.4 Estimating Pathway Crosstalk

Lastly, we used paths computed by LINKER for each pathway to estimate levels of crosstalk between signaling pathways. Pathway crosstalk occurs when the downstream effects of one pathway result from the stimulus of a different pathway. This may arise because different signaling pathways often use similar proteins and interactions to propagate signals through the cell. We estimated the amount of crosstalk between pathways by first computing the statistical significance (hypergeometric p-value) of the overlap between proteins from each pair of NetPath pathways, illustrated by the top heatmap in Figure 6.8. Since these values were computed directly from the NetPath protein sets, we considered this the known overlap between pairs of pathways. Note that the top heatmap in Figure 6.8 is naturally symmetric.

The middle heatmap in Figure 6.8 illustrates the statistical significance (hypergeometric p-value) of the overlap between the proteins in the top 250 pathways computed by LINKER (rows) and the known proteins in each NetPath pathway (columns). These values demonstrate crosstalk estimated by our computed paths with the existing pathways. Notice that the estimations for each pathway were highly enriched in proteins from their NetPath representations. This reiterates the findings from our precision-recall analyses that paths estimated by LINKER successfully recovered proteins from the corresponding NetPath pathways.

The bottom heatmap in Figure 6.8 provides the negative logarithm of the ratio of the estimated crosstalk p-values (middle) to the background crosstalk (top). Thus, large negative scores indicate crosstalk that has much stronger support from the known overlap. The diagonal elements naturally have more significant background overlap, since the sets are identical. Conversely, large positive scores denote pairs of pathways for which the estimated pathway overlap was much more significant than the background overlap between the same pair of NetPath pathways. A pair of pathways with a high ratio indicates pathway crosstalk that would not likely be discovered from analyzing the NetPath pathways alone.

As an example, LINKER predictions for BCR demonstrated a statistically significant overlap with the known TCR pathway; though not as significant, the predicted TCR pathway also highly overlapped with the known NetPath BCR pathway. The bottom heatmap in Figure 6.8 illustrates that the high overlap between predicted BCR and known TCR (and

vice versa) is not a surprising discovery since the known BCR and TCR NetPath pathways share many proteins.

Conversely, the paths predicted by LINKER for IL1 and the known TNFα pathway show one of the highest ratios of predicted to known overlaps, suggesting this pair of pathways utilizes crosstalk that is not readily apparent from the NetPath pathways. Experiments performed in rats have shown that intradermal injections of IL-1β resulted in localized increased production of TNF-α levels [18], suggesting that some downstream effects of the IL1 pathway (which is stimulated by IL-1β) upregulate the production of TNFα and potentially activate the TNFα pathway. Notably, the overlap between the predicted TNFα paths with the known IL1 pathway was comparatively low, suggesting that the crosstalk from TNFα to IL1 was less surprising than the crosstalk from IL1 to TNFα. The experimental evidence in rats supports our prediction that the IL1 pathway affects TNFα production [18]. Further experimental studies must be performed to test the other direction. Nonetheless, our asymmetric crosstalk scores between a pair of pathways provide intuitive estimations of which pathway may control another pathway through downstream effects.

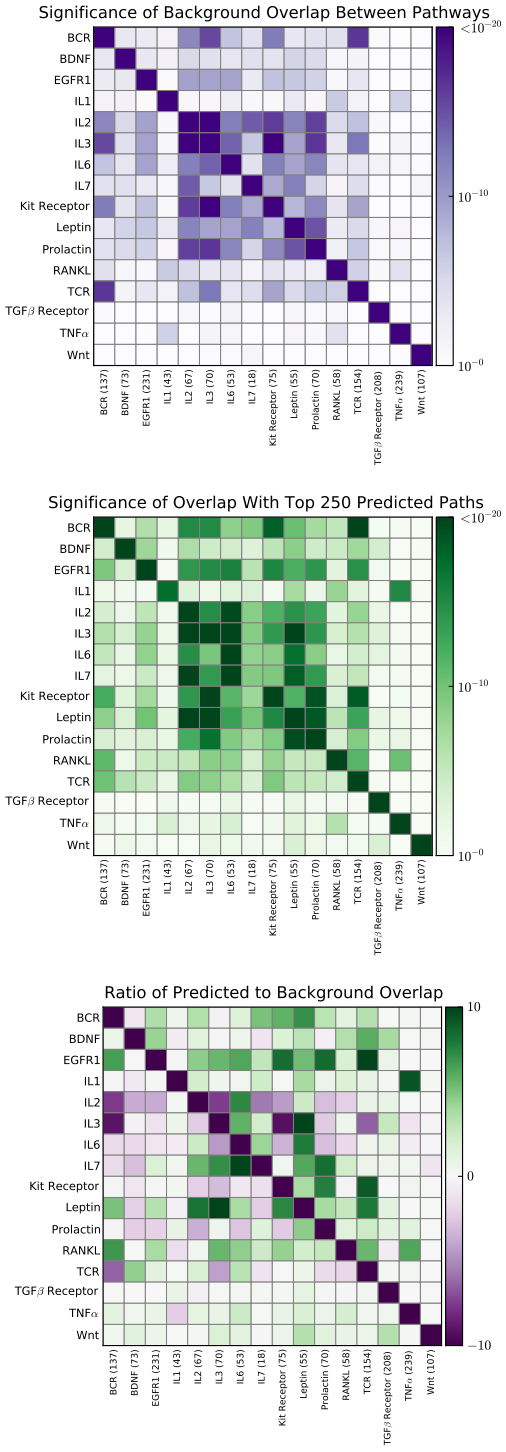


Figure 6.8: Estimated crosstalk between signaling pathways. Heatmaps indicating the statistical significance (hypergeometric p-value) of the (*top*) overlap between proteins in pairs of NetPath pathways and (*middle*) overlap between proteins participating in the top 250 paths computed by LINKER (rows) with proteins from each NetPath pathway (columns). (*bottom*) The negative logarithm of the ratio of the predicted pathway overlap to the background pathway overlap (i.e., the ratio of the p-values from the middle plot to those from the top plot). Parentheses indicate the number of proteins in the NetPath pathway.

6.5 Discussion

In this chapter, we addressed the problem of reconstructing signaling pathways from an interactome. We presented Linker as a novel approach to reconstruct signaling pathways, and we compared our method to several algorithms. We note that not all of the comparative methods were designed for the challenge of reconstructing signaling pathways. Specifically, PCST-based algorithms and approaches that compute small subnetworks are not ideal for recovering specific pathways, since the number of predicted nodes and edges is limited to the computed subnetwork. Linker supports an easy-to-interpret parameter, k (the number of shortest paths) upon whose increase the computed subnetworks expand smoothly. The ability to expand the computed subnetworks (by increasing k) coupled with the efficient run times in practice endows Linker to achieve far greater recall than many prominent competing methods such as ANAT, PCSF, and ResponseNet, which have been defined to compute succinct subnetworks that connect sources to targets. Our experimental collaborators previously found this property useful when we used Linker to propose extensions to a mathematical model of the cell cycle in budding yeast [97].

Our approach is space and time efficient in practice. We ran Linker on a Dell OptiPlex 990 Desktop with Intel Core i7-2600 CPUs at 3.4GHz; Linker used less than 0.5GB of memory during all executions. The EdgeFlux stage of Linker converged in a few seconds; thus, we focus only on the most time-consuming stage, KSP. Figure 6.9 illustrates the cumulative run times for Linker to enumerate the $k = 5000$ most probable paths for the NetPath pathways. The blue curve denotes the average time to compute k paths over all 16 pathways. Linker computed 1000 paths in under five minutes for every pathway. On average, Linker computed the 5000 paths in approximately 16 minutes for each pathway. The maximum and minimum times spent for a single pathway were 22 and 11 minutes, respectively.

Ideally, we seek accurate predictions that explain how the signal propagates from the cell surface receptors to downstream transcriptional changes in the nucleus. Thus, methods that rank nodes and edges in the network without considering their connectivity within the ranking are not ideal because top-ranked predictions are not guaranteed to contain a

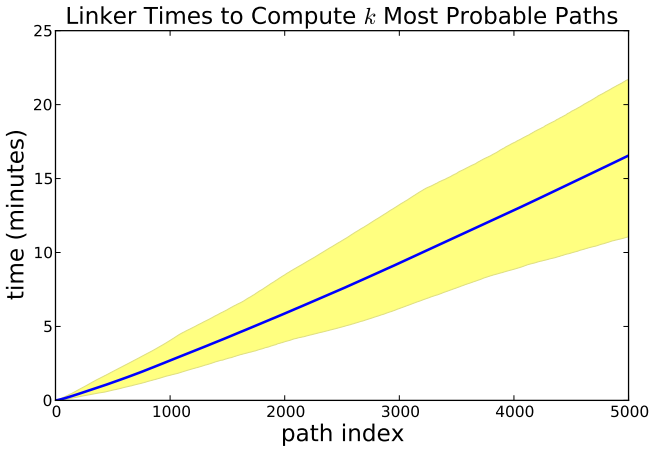


Figure 6.9: LINKER cumulative run times to compute $k = 5000$ most probable paths. The blue curve denotes the average cumulative run time for 16 NetPath pathways. The shaded region indicates the range of maximum and minimum run times over all 16 pathways.

path that links the receptors to downstream TRs. eQED and EDGEFLUX fall prey to this issue as the highest-scoring proteins and interactions are near the source proteins within the network; therefore, multiple paths linking receptors to TRs do not emerge until further down the ranked list. LINKER avoids this issue by explicitly computing paths and ranking interactions accordingly; thus, the first few predictions are guaranteed to suggest specific connections from receptors to TRs.

LINKER assigns a direction to each edge in the reconstructed pathways. We ignored the direction of predicted interactions for our precision-recall analyses since we wanted to evaluate LINKER's ability to reconstruct interacting pairs of proteins. However, predicting the orientation of signaling interactions is a timely problem that can provide clues to how a signal propagates through the cell [41].

We assumed a simplistic model of signaling pathways whereby the signal start and ends at two well-defined, disjoint sets of proteins. Signaling pathways often incorporate feedback mechanisms that can, for example, amplify the downstream effect of the signal. We did not address the issue of feedback in this work, but we envision extensions and alternative applications of LINKER that can reconstruct signaling pathways more faithfully by modeling feedback. One simple approach is to run LINKER in the reverse direction (from TRs to receptors) for each pathway. Such connections may highlight pathway feedback mechanisms.

Chapter 7

Conclusions

This thesis tackles three methodological gaps in network-based systems biology. We first introduce the concept of response networks and a variety of algorithms from the literature that are used to compute them. Response networks highlight subnetworks (of a much larger interactome) that include many proteins whose genes are significantly perturbed under a certain condition. The history of response networks and their applications provide relevant background for the three methodological gaps addressed in the remaining chapters.

Computational methods such as response networks often yield sets of proteins thought to be of interest to the condition that is being studied. Such protein sets are routinely analyzed through set-based functional enrichment methods. Chapter 3 addresses the first fundamental gap whereby traditional functional enrichment approaches ignore the presence or absence of interactions among the interesting set of proteins. We introduce a network-based functional enrichment algorithm that accounts for both the number of functionally annotated proteins in the interesting set and the connectivity of the proteins in the interactome. We demonstrate that this approach is a direct generalization of the one-sided version of Fisher's exact test, a commonly used set-based approach.

Response network algorithms intentionally compute small subnetworks to ease interpretability. As a result, such approaches may only highlight specific functions or pathways that are dysregulated by a particular disease. Chapter 4 addresses the second gap that small subnetworks do not uncover the spectrum of pathways that are perturbed by a condition. We present an approach that reconciles differential gene expression p-values with an underly-

ing interactome. After re-ranking the genes, we are able to highlight several highly-relevant biological functions that would be missed by analyzing small response networks or the gene expression data alone.

Lastly, Chapters 5 and 6 tackle the third gap of bridging top-down and bottom-up methodologies. We present a network-based top-down approach, LINKER, to automatically suggest extensions to existing bottom-up models, and we highlight the utility of LINKER through two different applications. In Chapter 5, we show that LINKER identifies relevant extensions to an existing model of the cell cycle in budding yeast. In Chapter 6, we utilize LINKER to reconstruct human signaling pathways. Domain experts have traditionally constructed mechanistic models of cellular processes and biological pathways through time-consuming manual curation. We demonstrate LINKER can expedite the this process by automatically identifying extensions to existing models.

Chapter 8

Bibliography

[1] D. Abdulrehman, P. T. Monteiro, M. C. Teixeira, N. P. Mira, A. B. Lourenco, S. C. dos Santos, T. R. Cabrito, A. P. Francisco, S. C. Madeira, R. S. Aires, A. L. Oliveira, I. Sa-Correia, and A. T. Freitas. YEASTRACT: providing a programmatic access to curated transcriptional regulatory associations in *Saccharomyces cerevisiae* through a web services interface. *Nucleic Acids Res.*, 39(Database issue):D136–140, Jan 2011.

[2] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular biology of the cell*. Garland Science, 2007.

[3] Markus Almen, Karl Nordstrom, Robert Fredriksson, and Helgi Schioth. Mapping the human membrane proteome: a majority of the human membrane proteins can be classified according to function and evolutionary origin. *BMC Biology*, 7(1):50+, 2009.

[4] U. Alon, M. G. Surette, N. Barkai, and S. Leibler. Robustness in bacterial chemotaxis. *Nature*, 397(6715):168–171, January 1999.

[5] A. Arkin, J. Ross, and H. H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected *Escherichia coli* cells. *Genetics*, 149(4):1633–1648, Aug 1998.

[6] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, 25(1):25–29, May 2000.

[7] M. Bailly-Bechet, C. Borgs, A. Braunstein, J. Chayes, A. Dagkessamanskaia, J. M. François, and R. Zecchina. Finding undetected protein associations in cell signaling by belief propagation. *Proceedings of the National Academy of Sciences*, 108(2):882–887, January 2011.

[8] Mukesh Bansal, Vincenzo Belcastro, Alberto Ambesi-Impiombato, and Diego di Bernardo. How to infer gene networks from expression profiles. *Mol Syst Biol*, 3, February 2007.

[9] Albert-Laszlo Barabasi, Natali Gulbahce, and Joseph Loscalzo. Network medicine: a network-based approach to human disease. *Nature Reviews Genetics*, 12(1):56–68, January 2011.

[10] Sebastian Bauer, Julien Gagneur, and Peter N. Robinson. GOing Bayesian: model-based gene set analysis of genome-scale data. *Nucleic Acids Research*, 38(11):3523–3532, 2010.

[11] J. Behrens, J. P. von Kries, M. Kuhl, L. Bruhn, D. Wedlich, R. Grosschedl, and W. Birchmeier. Functional interaction of β-catenin with the transcription factor LEF-1. *Nature*, 382(6592):638–642, Aug 1996.

[12] D. Beisser, G. W. Klau, T. Dandekar, T. Muller, and M. T. Dittrich. BioNet: an R-Package for the functional analysis of biological networks. *Bioinformatics*, 26(8):1129–30, 2010.

[13] Gabriel F Berriz, Oliver D King, Barbara Bryant, Chris Sander, and Frederick P Roth. Characterizing gene sets with FuncAssociate. *Bioinformatics*, 19(18):2502–4, 2003.

[14] Gabriel F. Berriz and Frederick P. Roth. The synergizer service for translating gene, protein and other biological identifiers. *Bioinformatics*, 24(19):2272–2273, October 2008.

[15] M. Bienz. β-Catenin: a pivot between cell adhesion and Wnt signalling. *Curr. Biol.*, 15(2):R64–67, Jan 2005.

[16] Bernd Bodenmiller, Stefanie Wanka, Claudine Kraft, Jorg Urban, David Campbell, Patrick G. Pedrioli, Bertran Gerrits, Paola Picotti, Henry Lam, Olga Vitek, Mi-Youn Brusniak, Bernd Roschitzki, Chao Zhang, Kevan M. Shokat, Ralph Schlapbach, Alejandro Colman-Lerner, Garry P. Nolan, Alexey I. Nesvizhskii, Matthias Peter, Robbie Loewith, Christian von Mering, and Ruedi Aebersold. Phosphoproteomic analysis reveals interconnected System-Wide responses to perturbations of kinases and phosphatases in yeast. *Sci. Signal.*, 3(153):rs4+, December 2010.

[17] F. H. Brembeck, T. Schwarz-Romond, J. Bakkers, S. Wilhelm, M. Hammerschmidt, and W. Birchmeier. Essential role of BCL9-2 in the switch between beta-catenin's adhesive and transcriptional functions. *Genes Dev.*, 18(18):2225–2230, Sep 2004.

[18] M. M. Campos, G. E. de Souza, N. D. Ricci, J. L. Pesquero, M. M. Teixeira, and J. B. Calixto. The role of migrating leukocytes in IL-1 beta-induced up-regulation of kinin B(1) receptors in rats. *Br. J. Pharmacol.*, 135(5):1107–1114, Mar 2002.

[19] Moses Charikar. Greedy approximation algorithms for finding dense components in a graph. In *Proceedings of 3rd International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 84–95. Springer, 2000.

[20] Moses Charikar, Chandra Chekuri, To-yat Cheung, Zuo Dai, Ashish Goel, Sudipto Guha, and Ming Li. Approximation algorithms for directed Steiner problems. *Journal of Algorithms*, 33(1):73–91, October 1999.

[21] Katherine C. Chen, Laurence Calzone, Attila Csikasz-Nagy, Frederick R. Cross, Bela Novak, and John J. Tyson. Integrative analysis of cell cycle control in budding yeast. *Molecular biology of the cell*, 15(8):3841–3862, August 2004.

[22] Hon N. Chua, Wing-Kin Sung, and Limsoon Wong. Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions. *Bioinformatics*, 22(13):1623–1630, July 2006.

[23] Fan Chung. The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences*, 104(50):19735–19740, December 2007.

[24] Sean R. Collins, Patrick Kemmeren, Xue-Chu Zhao, Jack F. Greenblatt, Forrest Spencer, Frank C. P. Holstege, Jonathan S. Weissman, and Nevan J. Krogan. Toward a comprehensive atlas of the physical interactome of *Saccharomyces cerevisiae*. *Mol Cell Proteomics*, 6(3):439–450, March 2007.

[25] Sean R. Collins, Kyle M. Miller, Nancy L. Maas, Assen Roguev, Jeffrey Fillingham, Clement S. Chu, Maya Schuldiner, Marinella Gebbia, Judith Recht, Michael Shales, Huiming Ding, Hong Xu, Junhong Han, Kristin Ingvarsdottir, Benjamin Cheng, Brenda Andrews, Charles Boone, Shelley L. Berger, Phil Hieter, Zhiguo Zhang, Grant W. Brown, C. James Ingles, Andrew Emili, C. David Allis, David P. Toczyski, Jonathan S. Weissman, Jack F. Greenblatt, and Nevan J. Krogan. Functional dissection of protein complexes involved in yeast chromosome biology using a genetic interaction map. *Nature*, 446(7137):806–810, April 2007.

[26] Markus W. Covert, Eric M. Knight, Jennifer L. Reed, Markus J. Herrgard, and Bernhard O. Palsson. Integrating high-throughput and computational data elucidates bacterial networks. *Nature*, 429(6987):92–96, May 2004.

[27] ILOG CPLEX. 11.0 users manual. *ILOG SA, Gentilly, France*, 2007.

[28] Suzanne Craft and G. Stennis Watson. Insulin and neurodegenerative disease: shared and specific mechanisms. *The Lancet Neurology*, 3(3):169–178, March 2004.

[29] Rahul C. Deo, Luke Hunter, Gregory D. Lewis, Guillaume Pare, Ramachandran S. Vasan, Daniel Chasman, Thomas J. Wang, Robert E. Gerszten, and Frederick P. Roth. Interpreting metabolomic profiles using unbiased pathway models. *PLoS Comput Biol*, 6(2):e1000692+, February 2010.

[30] Marcus T. Dittrich, Gunnar W. Klau, Andreas Rosenwald, Thomas Dandekar, and Tobias Muller. Identifying functional modules in protein-protein interaction networks: an integrated exact approach. *Bioinformatics*, 24(13):i223–231, July 2008.

[31] Michele K. Dougherty and Deborah K. Morrison. Unlocking the code of 14-3-3. *Journal of Cell Science*, 117(10):1875–1884, April 2004.

[32] Peter G. Doyle and S. J. Snell. *Random walks and electrical networks (carus mathematical monographs, no 22)*. Mathematical Assn of America, December 1984.

[33] Steffen Durinck. Pre-processing of microarray data and analysis of differential expression. *Methods in Molecular Biology (Clifton, N.J.)*, 452:89–110, 2008.

[34] S Fields and O Song. A novel genetic system to detect protein-protein interactions. *Nature*, 340(6230):245–6, 1989.

[35] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM J. Comput.*, 18(1):30–55, 1989.

[36] A. P. Gasch, M. Huang, S. Metzner, D. Botstein, S. J. Elledge, and P. O. Brown. Genomic expression responses to DNA-damaging agents and the regulatory role of the yeast ATR homolog Mec1p. *Mol Biol Cell*, 12(10):2987–3003, 2001.

[37] A P Gasch, P T Spellman, C M Kao, M B Eisen, G Storz, D Botstein, and P O Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Mol Biol Cell*, 11(12):4241–57, 2000.

[38] J. Gillis and P. Pavlidis. The impact of multifunctional genes on "guilt by association" analysis. *PLoS ONE*, 6(2):e17258, 2011.

[39] J. Gillis and P. Pavlidis. "Guilt by association" is the exception rather than the rule in gene networks. *PLoS Comput. Biol.*, 8(3):e1002444, 2012.

[40] Anthony Gitter, Miri Carmi, Naama Barkai, and Ziv Bar-Joseph. Linking the signaling cascades and dynamic regulatory networks controlling stress responses. *Genome Research*, 23(2):365–376, 2013.

[41] Anthony Gitter, Judith Klein-Seetharaman, Anupam Gupta, and Ziv Bar-Joseph. Discovering pathways by orienting edges in protein interaction networks. *Nucleic Acids Research*, 39(4):e22, March 2011.

[42] Joana P. Gonçalves, Alexandre P. Francisco, Nuno P. Mira, Miguel C. Teixeira, Isabel Sá-Correia, Arlindo L. Oliveira, and Sara C. Madeira. TFRank: network-based prioritization of regulatory associations underlying transcriptional responses. *Bioinformatics*, 27(22):3149–3157, November 2011.

[43] S. Grossmann, S. Bauer, P. N. Robinson, and M. Vingron. Improved Detection of Overrepresentation of Gene-Ontology Annotations with Parent-Child Analysis. *Bioinformatics*, 2007.

[44] Daniel Hanisch, Alexander Zien, Ralf Zimmer, and Thomas Lengauer. Co-clustering of biological networks and gene expression data. *Bioinformatics*, 18(suppl_1):S145–154, July 2002.

[45] C. T. Harbison, D. B. Gordon, T. I. Lee, N. J. Rinaldi, K. D. Macisaac, T. W. Danford, N. M. Hannett, J. B. Tagne, D. B. Reynolds, J. Yoo, E. G. Jennings, J. Zeitlinger, D. K. Pokholok, M. Kellis, P. A. Rolfe, K. T. Takusagawa, E. S. Lander, D. K. Gifford, E. Fraenkel, and R. A. Young. Transcriptional regulatory code of a eukaryotic genome. *Nature*, 431(7004):99–104, Sep 2004.

[46] Tony JC Harris and Ulrich Tepass. Adherens junctions: from molecules to morphogenesis. *Nature Reviews Molecular Cell Biology*, 11(7):502–514, 2010.

[47] L. M. Heltemes-Harris, M. J. Willette, K. B. Vang, and M. A. Farrar. The role of STAT5 in the development, function, and transformation of B and T lymphocytes. *Ann. N. Y. Acad. Sci.*, 1217:18–31, Jan 2011.

[48] Nicola J. Hewitt, María J. Gómez Lechón, J. Brian Houston, David Hallifax, Hayley S. Brown, Patrick Maurel, J. Gerald Kenna, Lena Gustavsson, Christina Lohmann, Christian Skonberg, Andre Guillouzo, Gregor Tuschl, Albert P. Li, Edward LeCluyse, Geny M. M. Groothuis, and Jan G. Hengstler. Primary hepatocytes: Current understanding of the regulation of metabolic enzymes and transporter proteins, and pharmaceutical practice for the use of hepatocytes in metabolism, enzyme induction, transporter, clearance, and hepatotoxicity studies. *Drug Metabolism Reviews*, 39(1):159–234, January 2007.

[49] Shao-shan C. Huang and Ernest Fraenkel. Integrating proteomic, transcriptional, and interactome data reveals hidden components of signaling and regulatory networks. *Sci. Signal.*, 2(81):ra40+, July 2009.

[50] Trey Ideker, Owen Ozier, Benno Schwikowski, and Andrew F Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18 Suppl 1:S233–40, 2002.

[51] Trey Ideker and Roded Sharan. Protein networks in disease. *Genome Research*, 18(4):644–652, 2008.

[52] Yong-Ri Jin and Jeong K. Yoon. The R-spondin family of proteins: Emerging regulators of Wnt signaling. *The International Journal of Biochemistry & Cell Biology*, 44(12):2278–2287, December 2012.

[53] Marc Johannes, Jan C. Brase, Holger Fröhlich, Stephan Gade, Mathias Gehrmann, Maria Fälth, Holger Sültmann, and Tim Beißbarth. Integration of pathway knowledge into a reweighted recursive feature elimination approach for risk stratification of cancer patients. *Bioinformatics*, 26(17):2136–2144, September 2010.

[54] Paul Jorgensen, Joy L. Nishikawa, Bobby-Joe Breitkreutz, and Mike Tyers. Systematic identification of pathways that couple cell growth and division in yeast. *Science*, 297(5580):395–400, July 2002.

[55] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G.R. Gopinath, G.R. Wu, L. Matthews, S. Lewis, E. Birney, and L. Stein. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Research*, 33(Database issue):D428–32, 2005.

[56] A. R. Joyce and B. O. Palsson. The model organism as a system: integrating 'omics' data sets. *Nat Rev Mol Cell Biol*, 7(3):198–210, 2006.

[57] K. Kandasamy, S. S. Mohan, R. Raju, S. Keerthikumar, G. S. Kumar, A. K. Venugopal, D. Telikicherla, J. D. Navarro, S. Mathivanan, C. Pecquet, S. K. Gollapudi, S. G. Tattikota, S. Mohan, H. Padhukasahasram, Y. Subbannayya, R. Goel, H. K. Jacob, J. Zhong, R. Sekhar, V. Nanjappa, L. Balakrishnan, R. Subbaiah, Y. L. Ramachandra, B. A. Rahiman, T. S. Prasad, J. X. Lin, J. C. Houtman, S. Desiderio, J. C. Renauld, S. N. Constantinescu, O. Ohara, T. Hirano, M. Kubo, S. Singh, P. Khatri, S. Draghici, G. D. Bader, C. Sander, W. J. Leonard, and A. Pandey. NetPath: a public resource of curated signal transduction pathways. *Genome Biol*, 11(1):R3, 2010.

[58] M. Kanehisa, M. Araki, S. Goto, M. Hattori, M. Hirakawa, M. Itoh, T. Katayama, S. Kawashima, S. Okuda, T. Tokimatsu, and Y. Yamanishi. KEGG for linking genomes to life and the environment. *Nucleic Acids Res.*, 36(Database issue):D480–484, Jan 2008.

[59] M. Kanehisa, S. Goto, M. Hattori, K. F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Research*, 34(Database issue):D354–7, 2006.

[60] M. Kanehisa, S. Goto, Y. Sato, M. Furumichi, and M. Tanabe. KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Res.*, 40(Database issue):D109–114, Jan 2012.

[61] Andreas Keller, Christina Backes, Andreas Gerasch, Michael Kaufmann, Oliver Kohlbacher, Eckart Meese, and Hans-Peter Lenhof. A novel algorithm for detecting differentially regulated paths based on gene set enrichment analysis. *Bioinformatics*, 25(21):2787–2794, November 2009.

[62] R Kelley and T Ideker. Systematic interpretation of genetic interactions using protein networks. *Nat Biotechnol*, 23(5):561–6, May 2005.

[63] Purvesh Khatri, Marina Sirota, and Atul J Butte. Ten years of pathway analysis: current approaches and outstanding challenges. *PLoS computational biology*, 8(2):e1002375, 2012.

[64] Kyung-Ah Kim, Marie Wagle, Karolyn Tran, Xiaoming Zhan, Melissa A. Dixon, Shouchun Liu, Delphine Gros, Wouter Korver, Shirlee Yonkovich, Nenad Tomasevic, Minke Binnerts, and Arie Abo. R-spondin family members regulate the Wnt pathway by a common mechanism. *Molecular Biology of the Cell*, 19(6):2588–2596, June 2008.

[65] Seon Y. Kim and David Volsky. PAGE: Parametric Analysis of Gene Set Enrichment. *BMC Bioinformatics*, 6(1):144+, 2005.

[66] Y. Kim, C. D. Lasher, L. M. Milford, T. M. Murali, and P. Rajagopalan. A Comparative Study of Genome-Wide Transcriptional Profiles of Primary Hepatocytes in Collagen Sandwich and Monolayer Cultures. *Tissue Eng Part C Methods*, 2010.

[67] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May 1983.

[68] Sebastian Köhler, Sebastian Bauer, Denise Horn, and Peter N. Robinson. Walking the interactome for prioritization of candidate disease genes. *American journal of human genetics*, 82(4):949–958, April 2008.

[69] K. Komurov, M. A. White, and P. T. Ram. Use of data-biased random walks on graphs for the retrieval of context-specific networks from genomic data. *PLoS Comput Biol*, 6(8), 2010.

[70] S.E. Lee, L.M. Frenz, N.J. Wells, A.L. Johnson, and L.H. Johnston. Order of function of the budding-yeast mitotic exit-network proteins Tem1, Cdc15, Mob1, Dbf2, and Cdc5. *Current Biology*, 11(10):784–788, 2001.

[71] Tong Ihn Lee, Nicola J. Rinaldi, Francois Robert, Duncan T. Odom, Ziv Bar-Joseph, Georg K. Gerber, Nancy M. Hannett, Christopher T. Harbison, Craig M. Thompson, Itamar Simon, Julia Zeitlinger, Ezra G. Jennings, Heather L. Murray, D. Benjamin Gordon, Bing Ren, John J. Wyrick, Jean-Bosco Tagne, Thomas L. Volkert, Ernest Fraenkel, David K. Gifford, and Richard A. Young. Transcriptional Regulatory Networks in *Saccharomyces cerevisiae*. *Science*, 298(5594):799–804, 2002.

[72] Celine Lefebvre, Wei K. Lim, Katia Basso, Riccardo D. Favera, and Andrea Califano. A context-specific network of protein-DNA and protein-protein interactions reveals new regulatory motifs in human b cells. In *Proceedings of the Joint 2006 satellite conference on Systems Biology and Computational Proteomics*, pages 42–56, Berlin, Heidelberg, 2007. Springer-Verlag.

[73] A. Liberzon, A. Subramanian, R. Pinchback, H. Thorvaldsdottir, P. Tamayo, and J. P. Mesirov. Molecular signatures database (MSigDB) 3.0. *Bioinformatics*, 27(12):1739–1740, Jun 2011.

[74] Manway Liu, Arthur Liberzon, Sek W. Kong, Weil R. Lai, Peter J. Park, Isaac S. Kohane, and Simon Kasif. Network-based analysis of affected biological processes in type 2 diabetes models. *PLoS Genet*, 3(6):e96+, June 2007.

[75] I. Ljubić, R. Weiskircher, U. Pferschy, G.W. Klau, P. Mutzel, and M. Fischetti. An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. *Mathematical Programming*, 105(2):427–449, 2006.

[76] V. D. Longo, E. B. Gralla, and J. S. Valentine. Superoxide dismutase activity is essential for stationary phase survival in *Saccharomyces cerevisiae*. Mitochondrial production of toxic oxygen species *in vivo*. *J. Biol. Chem.*, 271(21):12275–12280, May 1996.

[77] Yong Lu, Roni Rosenfeld, Itamar Simon, Gerard J. Nau, and Ziv Bar-Joseph. A probabilistic generative model for GO enrichment analysis. *Nucl. Acids Res.*, 36(17):e109+, October 2008.

[78] S. Malin, S. McManus, and M. Busslinger. STAT5 in B cell development and leukemia. *Curr. Opin. Immunol.*, 22(2):168–176, Apr 2010.

[79] Adam A. Margolin, Kai Wang, Wei K. Lim, Manjunath Kustagi, Ilya Nemenman, and Andrea Califano. Reverse engineering cellular networks. *Nature Protocols*, 1(2):662–671, June 2006.

[80] Florian Markowetz and Rainer Spang. Inferring cellular networks - a review. *BMC Bioinformatics*, 8(Suppl 6):S5, 2007.

[81] L. Matthews, G. Gopinath, M. Gillespie, M. Caudy, D. Croft, B. de Bono, P. Garapati, J. Hemish, H. Hermjakob, B. Jassal, A. Kanapin, S. Lewis, S. Mahajan, B. May, E. Schmidt, I. Vastrik, G. Wu, E. Birney, L. Stein, and P. D'Eustachio. Reactome knowledgebase of human biological pathways and processes. *Nucleic Acids Research*, 37(Database issue):D619–22, 2009.

[82] R. Milo, N. Kashtan, S. Itzkovitz, MEJ Newman, and U. Alon. On the uniform generation of random graphs with prescribed degree sequences. *Arxiv preprint cond-mat/0312028*, 2003.

[83] R Milo, S Shen-Orr, S Itzkovitz, N Kashtan, D Chklovskii, and U Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–7, 2002.

[84] Fazilat F. Mohammed and Rama Khokha. Thinking outside the cell: proteases regulate hepatocyte division. *Trends in Cell Biology*, 15(10):555–563, October 2005.

[85] S. Mostafavi, D. Ray, D. Warde-Farley, C. Grouios, and Q. Morris. GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biol*, 9 Suppl 1:S4, 2008.

[86] T. M. Murali, Matthew D. Dyer, David Badger, Brett M. Tyler, and Michael G. Katze. Network-Based prediction and analysis of HIV dependency factors. *PLoS Comput Biol*, 7(9):e1002164+, September 2011.

[87] T. M. Murali and Corban G. Rivera. Network legos: Building blocks of cellular wiring diagrams. *Journal of Computational Biology*, 15(7):829–844, 2008.

[88] Chad Myers, Daniel Barrett, Matthew Hibbs, Curtis Huttenhower, and Olga Troyanskaya. Finding function: evaluation methods for functional genomic data. *BMC Genomics*, 7(1):187+, July 2006.

[89] Saket Navlakha and Carl Kingsford. The power of protein interaction networks for associating genes with diseases. *Bioinformatics (Oxford, England)*, 26(8):1057–1063, April 2010.

[90] NCI-Nature Pathway Interaction Database. http://pid.nci.nih.gov.

[91] Christof Niehrs. The complex world of WNT receptor signalling. *Nat Rev Mol Cell Biol*, 13(12):767–779, December 2012.

[92] Daniela Nitsch, Joana Goncalves, Fabian Ojeda, Bart de Moor, and Yves Moreau. Candidate gene prioritization by network analysis of differential expression using machine learning approaches. *BMC Bioinformatics*, 11(1):460+, September 2010.

[93] Roel Nusse. Wnt signaling. *Cold Spring Harbor Perspectives in Biology*, 4(5), 2012.

[94] O. Ourfali, T. Shlomi, T. Ideker, E. Ruppin, and R. Sharan. SPINE: a framework for signaling-regulatory pathway inference from cause-effect experiments. *Bioinformatics*, 23(13):i359–66, 2007.

[95] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999.

[96] Arnon Paz, Zippora Brownstein, Yaara Ber, Shani Bialik, Eyal David, Dorit Sagir, Igor Ulitsky, Ran Elkon, Adi Kimchi, Karen B. Avraham, Yosef Shiloh, and Ron Shamir. SPIKE: a database of highly curated human signaling pathways. *Nucleic Acids Research*, 39(suppl 1):D793–D799, January 2011.

[97] C. L. Poirel, R. R. Rodrigues, K. C. Chen, J. J. Tyson, and T. M. Murali. Top-down network analysis to drive bottom-up modeling of physiological processes. *J. Comput. Biol.*, 20(5):409–418, May 2013.

[98] Christopher L. Poirel, Clifford C. Owens III, and T. M. Murali. Network-based functional enrichment. *BMC Bioinformatics*, 12(Suppl 13):S14, 2011.

[99] Christopher L Poirel, Ahsanur Rahman, Richard R Rodrigues, Arjun Krishnan, Jacqueline R Addesa, and TM Murali. Reconciling differential gene expression data with molecular interaction networks. *Bioinformatics*, 29(5):622–629, 2013.

[100] Stan Pounds and Stephan W. Morris. Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partitioning the empirical distribution of p-values. *Bioinformatics*, 19(10):1236–1242, July 2003.

[101] Yu Q. Qiu, Shihua Zhang, Xiang S. Zhang, and Luonan Chen. Detecting disease associated modules and prioritizing active genes based on high throughput data. *BMC Bioinformatics*, 11(1):26+, 2010.

[102] Timothy Ravasi, Harukazu Suzuki, Carlo V. Cannistraci, Shintaro Katayama, Vladimir B. Bajic, Kai Tan, Altuna Akalin, Sebastian Schmeier, Mutsumi Kanamori-Katayama, Nicolas Bertin, Piero Carninci, Carsten O. Daub, Alistair R. R. Forrest, Julian Gough, Sean Grimmond, Jung-Hoon Han, Takehiro Hashimoto, Winston Hide, Oliver Hofmann, Atanas Kamburov, Mandeep Kaur, Hideya Kawaji, Atsutaka Kubosaki, Timo Lassmann, Erik van Nimwegen, Cameron R. MacPherson, Chihiro Ogawa, Aleksandar Radovanovic, Ariel Schwartz, Rohan D. Teasdale, Jesper Tegnér, Boris Lenhard, Sarah A. Teichmann, Takahiro Arakawa, Noriko Ninomiya, Kayoko Murakami, Michihira Tagami, Shiro Fukuda, Kengo Imamura, Chikatoshi Kai, Ryoko Ishihara, Yayoi Kitazume, Jun Kawai, David A. Hume, Trey Ideker, and Yoshihide Hayashizaki. An atlas of combinatorial transcriptional regulation in mouse and man. *Cell*, 140(5):744–752, March 2010.

[103] T. Reguly, A. Breitkreutz, L. Boucher, B. J. Breitkreutz, G. C. Hon, C. L. Myers, A. Parsons, H. Friesen, R. Oughtred, A. Tong, C. Stark, Y. Ho, D. Botstein, B. Andrews, C. Boone, O. G. Troyanskya, T. Ideker, K. Dolinski, N. N. Batada, and M. Tyers. Comprehensive curation and analysis of global interaction networks in *Saccharomyces cerevisiae*. *J. Biol.*, 5(4):11, 2006.

[104] Hyeon-Su S. Ro, Sukgil Song, and Kyung S. Lee. Bfa1 can regulate Tem1 function independently of Bub2 in the mitotic exit network of *Saccharomyces cerevisiae*. *Proceedings of the National Academy of Sciences*, 99(8):5436–5441, April 2002.

[105] P.J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

[106] J.F. Rual, K. Venkatesan, T. Hao, T. Hirozane-Kishikawa, A. Dricot, N. Li, G.F. Berriz, F.D. Gibbons, M. Dreze, N. Ayivi-Guedehoussou, N. Klitgord, C. Simon, M. Boxem, S. Milstein, J. Rosenberg, D.S. Goldberg, L.V. Zhang, S.L. Wong, G. Franklin, S. Li, J.S. Albala, J. Lim, C. Fraughton, E. Llamosas, S. Cevik, C. Bex, P. Lamesch, R.S. Sikorski, J. Vandenhaute, H.Y. Zoghbi, A. Smolyar, S. Bosak, R. Sequerra, L. Doucette-Stamm, M.E. Cusick, D.E. Hill, F.P. Roth, and M. Vidal. Towards a proteome-scale map of the human protein-protein interaction network. *Nature*, 437(7062):1173–8, 2005.

[107] Andreas Ruepp, Barbara Brauner, Irmtraud Dunger-Kaltenbach, Goar Frishman, Corinna Montrone, Michael Stransky, Brigitte Waegele, Thorsten Schmidt, Octave Noubibou N. Doudieu, Volker Stümpflen, and H. Werner Mewes. CORUM: the comprehensive resource of mammalian protein complexes. *Nucleic Acids Research*, 36(Database issue):D646–650, January 2008.

[108] Lukasz Salwinski, Christopher S Miller, Adam J Smith, Frank K Pettit, James U Bowie, and David Eisenberg. The Database of Interacting Proteins: 2004 update. *Nucleic Acids Research*, 32 Database issue:D449–51, 2004.

[109] C. F. Schaefer, K. Anthony, S. Krupa, J. Buchoff, M. Day, T. Hannay, and K. H. Buetow. PID: the Pathway Interaction Database. *Nucleic Acids Research*, 37(Database issue):D674–D679, 2009.

[110] J. Scott, T. Ideker, R. M. Karp, and R. Sharan. Efficient algorithms for detecting signaling pathways in protein interaction networks. *J. Comput. Biol.*, 13(2):133–144, Mar 2006.

[111] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–504, 2003.

[112] Sara Sharifpoor, Alex N. Ba, Ji Y. Young, Dewald van Dyk, Helena Friesen, Alison Douglas, Christoph Kurat, Yolanda Chong, Karen Founk, Alan Moses, and Brenda

Andrews. A quantitative literature-curated gold standard for kinase-substrate pairs. *Genome Biology*, 12(4):R39+, April 2011.

[113] Yu-Keng K. Shih and Srinivasan Parthasarathy. A single source k-shortest paths algorithm to infer regulatory pathways in a gene network. *Bioinformatics (Oxford, England)*, 28(12):i49–i58, June 2012.

[114] G. K. Smyth. limma: Linear Models for Microarray Data Bioinformatics and Computational Biology Solutions Using R and Bioconductor. In Robert Gentleman, Vincent J. Carey, Wolfgang Huber, Rafael A. Irizarry, and Sandrine Dudoit, editors, *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, Statistics for Biology and Health, chapter 23, pages 397–420. Springer New York, New York, 2005.

[115] Laura Soucek, Jonathan Whitfield, Carla P. Martins, Andrew J. Finch, Daniel J. Murphy, Nicole M. Sodir, Anthony N. Karnezis, Lamorna B. Swigart, Sergio Nasi, and Gerard I. Evan. Modelling Myc inhibition as a cancer therapy. *Nature*, 455(7213):679–683, August 2008.

[116] Chris Stark, Bobby-Joe J. Breitkreutz, Andrew Chatr-Aryamontri, Lorrie Boucher, Rose Oughtred, Michael S. Livstone, Julie Nixon, Kimberly Van Auken, Xiaodong Wang, Xiaoqi Shi, Teresa Reguly, Jennifer M. Rust, Andrew Winter, Kara Dolinski, and Mike Tyers. The BioGRID interaction database: 2011 update. *Nucleic Acids Research*, 39(Database issue):D698–D704, 2011.

[117] Martin Steffen, Allegra Petti, John Aach, Patrik D'haeseleer, and George Church. Automated modelling of signal transduction networks. *BMC Bioinformatics*, 3(1):34+, November 2002.

[118] A. Subramanian, P. Tamayo, V.K. Mootha, S. Mukherjee, B.L. Ebert, M.A. Gillette, A. Paulovich, S.L. Pomeroy, T.R. Golub, E.S. Lander, and J.P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 2005.

[119] S. Suthram, A. Beyer, R. M. Karp, Y. Eldar, and T. Ideker. eQED: an efficient method for interpreting eQTL associations using protein networks. *Mol. Syst. Biol.*, 4:162, 2008.

[120] Silpa Suthram, Joel T. Dudley, Annie P. Chiang, Rong Chen, Trevor J. Hastie, and Atul J. Butte. Network-based elucidation of human disease similarities reveals common functional modules enriched for pluripotent drug targets. *PLoS Comput Biol*, 6(2):e1000662+, February 2010.

[121] 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 von Mering. The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. *Nucleic Acids Research*, 39(Database issue):D561–D568, January 2011.

[122] V. G. Tarcea, T. Weymouth, A. Ade, A. Bookvich, J. Gao, V. Mahavisno, Z. Wright, A. Chapman, M. Jayapandian, A. Ozgur, Y. Tian, J. Cavalcoli, B. Mirel, J. Patel, D. Radev, B. Athey, D. States, and H. V. Jagadish. Michigan molecular interactions r2: from interacting proteins to pathways. *Nucleic Acids Research*, 37(Database issue):D642–6, 2009.

[123] Amy Hin Yan Tong, Guillaume Lesage, Gary D Bader, Huiming Ding, Hong Xu, Xiaofeng Xin, James Young, Gabriel F Berriz, Renee L Brost, Michael Chang, YiQun Chen, Xin Cheng, Gordon Chua, Helena Friesen, Debra S Goldberg, Jennifer Haynes, Christine Humphries, Grace He, Shamiza Hussein, Lizhu Ke, Nevan Krogan, Zhijian Li, Joshua N Levinson, Hong Lu, Patrice Ménard, Christella Munyana, Ainslie B Parsons, Owen Ryan, Raffi Tonikian, Tania Roberts, Anne-Marie Sdicu, Jesse Shapiro, Bilal Sheikh, Bernhard Suter, Sharyl L Wong, Lan V Zhang, Hongwei Zhu, Christopher G Burd, Sean Munro, Chris Sander, Jasper Rine, Jack Greenblatt, Matthias Peter, Anthony Bretscher, Graham Bell, Frederick P Roth, Grant W Brown, Brenda Andrews, Howard Bussey, and Charles Boone. Global mapping of the yeast genetic interaction network. *Science*, 303(5659):808–13, 2004.

[124] N. Tuncbag, A. Braunstein, A. Pagnani, S. S. Huang, J. Chayes, C. Borgs, R. Zecchina, and E. Fraenkel. Simultaneous reconstruction of multiple signaling pathways via the prize-collecting Steiner forest problem. *J. Comput. Biol.*, 20(2):124–136, Feb 2013.

[125] 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, Feb 2000.

[126] I. Ulitsky and R. Shamir. Identification of functional modules using network topology and high-throughput data. *BMC Syst Biol*, 1:8, 2007.

[127] Igor Ulitsky, Akshay Krishnamurthy, Richard M. Karp, and Ron Shamir. DEGAS: De Novo Discovery of Dysregulated Pathways in Human Diseases. *PLoS ONE*, 5(10):e13367, 2010.

[128] Igor Ulitsky and Ron Shamir. Identifying functional modules using expression profiles and confidence-scored protein interactions. *Bioinformatics*, 25(9):1158–1164, May 2009.

[129] F. Vandin, E. Upfal, and B. J. Raphael. Algorithms for detecting significantly mutated pathways in cancer. *J. Comput. Biol.*, 18(3):507–522, Mar 2011.

[130] Oron Vanunu, Oded Magger, Eytan Ruppin, Tomer Shlomi, and Roded Sharan. Associating genes and protein complexes with disease via network propagation. *PLoS Comput Biol*, 6(1):e1000641+, January 2010.

[131] Juan M. Vaquerizas, Sarah K. Kummerfeld, Sarah A. Teichmann, and Nicholas M. Luscombe. A census of human transcription factors: function, expression and evolution. *Nat Rev Genet*, 10(4):252–263, April 2009.

[132] Jean P. Vert and Minoru Kanehisa. Extracting active pathways from gene expression data. *Bioinformatics*, 19(suppl 2):ii238–ii244, September 2003.

[133] Arunachalam Vinayagam, Ulrich Stelzl, Raphaele Foulle, Stephanie Plassmann, Martina Zenkner, Jan Timm, Heike E. Assmus, Miguel A. Andrade-Navarro, and Erich E. Wanker. A directed protein interaction network for investigating intracellular signal transduction. *Sci. Signal.*, 4(189):rs8+, September 2011.

[134] Kai Wang, Masumichi Saito, Brygida C. Bisikirska, Mariano J. Alvarez, Wei K. Lim, Presha Rajbhandari, Qiong Shen, Ilya Nemenman, Katia Basso, Adam A. Margolin, Ulf Klein, Riccardo Dalla-Favera, and Andrea Califano. Genome-wide identification of post-translational modulators of transcription factor activity in human B cells. *Nature Biotechnology*, 27(9):829–837, 2009.

[135] Erik S. Welf and Jason M. Haugh. Signaling pathways that control cell migration: models and analysis. *Wiley interdisciplinary reviews. Systems biology and medicine*, 3(2):231–240, March 2011.

[136] Christof Winter, Glen Kristiansen, Stephan Kersting, Janine Roy, Daniela Aust, Thomas Knösel, Petra Rümmele, Beatrix Jahnke, Vera Hentrich, Felix Rückert, Marco Niedergethmann, Wilko Weichert, Marcus Bahra, Hans J. Schlitt, Utz Settmacher, Helmut Friess, Markus Büchler, Hans-Detlev Saeger, Michael Schroeder, Christian Pilarsky, and Robert Grützmann. Google goes cancer: Improving outcome prediction for cancer patients by network-based ranking of marker genes. *PLoS Comput Biol*, 8(5):e1002511+, May 2012.

[137] C. T. Workman, H. C. Mak, S. McCuine, J. B. Tagne, M. Agarwal, O. Ozier, T. J. Begley, L. D. Samson, and T. Ideker. A systems approach to mapping DNA damage response pathways. *Science*, 312(5776):1054–1059, May 2006.

[138] Haixuan Yang, Irwin King, and Michael R. Lyu. DiffusionRank: a possible penicillin for web spamming. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '07, pages 431–438, New York, NY, USA, 2007. ACM.

[139] C. H. Yeang, T. Ideker, and T. Jaakkola. Physical network models. *J Comput Biol*, 11(2-3):243–262, 2004.

[140] Chen H. Yeang, H. Craig Mak, Scott McCuine, Christopher Workman, Tommi Jaakkola, and Trey Ideker. Validation and refinement of gene-regulatory pathways on a network of physical interactions. *Genome Biology*, 6(7):R62+, 2005.

[141] Esti Yeger-Lotem, Laura Riva, Linhui Julie J. Su, Aaron D. Gitler, Anil G. Cashikar, Oliver D. King, Pavan K. Auluck, Melissa L. Geddie, Julie S. Valastyan, David R.

Karger, Susan Lindquist, and Ernest Fraenkel. Bridging high-throughput genetic and transcriptional data reveals cellular responses to alpha-synuclein toxicity. *Nature Genetics*, 41(3):316–323, March 2009.

[142] Jin Y. Yen. Finding the k shortest loopless paths in a network. *Management Science*, 17(11):712–716, 1971.

[143] Nir Yosef, Lior Ungar, Einat Zalckvar, Adi Kimchi, Martin Kupiec, Eytan Ruppin, and Roded Sharan. Toward accurate reconstruction of functional protein networks. *Molecular Systems Biology*, 5, March 2009.

[144] Nir Yosef, Einat Zalckvar, Assaf D. Rubinstein, Max Homilius, Nir Atias, Liram Vardi, Igor Berman, Hadas Zur, Adi Kimchi, Eytan Ruppin, and Roded Sharan. ANAT: A tool for constructing and analyzing functional protein networks. *Sci. Signal.*, 4(196):pl1+, October 2011.

[145] Mohammed Javeed Zaki and Ching-Jui Hsiao. Efficient algorithms for mining closed itemsets and their lattice structure. *IEEE Trans. Knowl. Data Eng.*, 17(4):462–478, 2005.

[146] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. *Advances in Neural Information Processing Systems*, 16:321–328, 2004.

[147] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *The Twentieth International Conference on Machine Learning, August 21-24, 2003, Washington, DC USA*, pages 912–919, 2003.