Algorithms for analyzing and interrogating protein interaction networks

Eric Banks

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Computer Science
Adviser: Mona Singh

April 2009

Abstract

High-throughput experimental and computational approaches are facilitating the collection of large-scale biological networks, consisting of proteins and the interactions among them, for a growing number of species. With appropriate computational analysis and experimental work the potential exists for uncovering the organizational principles of the cell and, consequently, protein functions and pathways, which are still largely unidentified. In this work, we introduce a novel framework for analyzing protein interaction networks in order to uncover organizational units corresponding to recurring means with which diverse biological processes are carried out. We formalize recurring patterns of interaction among different types of proteins using "network schemas"; network schemas specify descriptions of proteins and the topology of interactions among them.

In the first part of this thesis, we develop algorithms for systematically uncovering recurring, over-represented schemas in physical interaction networks and apply these methods to the *S. cerevisiae* interactome, identifying hundreds of such organizational units of varying complexity. We establish the functional importance of these schemas by showing that they correspond to functionally cohesive sets of proteins, are enriched in the frequency with which they have instances in the *H. sapiens* interactome, and are useful for predicting protein function. In the second part of this thesis, we introduce NetGrep, a system for searching protein interaction networks for matches to more general network schemas. NetGrep provides an advanced graphical interface for specifying schemas and fast algorithms for extracting their matches.

Acknowledgements

Chapter 2 is joint work with Elena Nabieva, Bernard Chazelle, and Mona Singh. It appeared in PLOS Computational Biology with the following reference:
Banks E, Nabieva E, Chazelle B, Singh M (2008) Organization of Physical Interactomes as Uncovered by Network Schemas. PLoS Comput Biol 4(10): e1000203 (doi:10.1371/journal.pcbi.1000203).
I would also like to thank Moses Charikar and David Blei for their helpful input about this project.

Chapter 3 is joint work with Elena Nabieva, Ryan Peterson, and Mona Singh. It appeared in Genome Biology with the following reference:
Banks E, Nabieva E, Peterson R, Singh M (2008) NetGrep: fast network schema searches in interactomes. Genome Biology 2008, 9:R138 (doi:10.1186/gb-2008-9-9-r138).

I would like to thank the members of my thesis committee, especially the readers Bernard Chazelle and Olga Troyanskaya, as well as the non-readers David Blei and Tom Funkhouser.

I was supported by Princeton University and by the Quantitative and Computational Biology Program NIH grant T32 HG003284.

I would like to thank the members of the Singh group, past and present, for discussions about this work as it was being developed. Extra thanks go to Elena Zaslavsky, Robert Osada, Tony Capra, and Anton Persikov. Elena Nabieva, my collaborator and 'partner in crime' for the majority of this work, gets my deep appreciation for her help and hard work.

I would like to thank my wife, Tamar, for her support, encouragement, and helpful instruction in the world of molecular biology throughout my years in graduate school. Thanks also go to my parents for instilling in me a thirst for knowledge and for supporting my lazy behind.

Finally, and most importantly, I am deeply indebted to my adviser, Mona Singh. Besides being an excellent instructor and mentor, she supported someone whose personal situation must have been terribly frustrating at times, and for that I am eternally thankful.

Contents

Chapter 1

Introduction

High-throughput experimental and computational approaches to characterize proteins and their interactions have resulted in large-scale biological networks for many organisms. Although biologists are accumulating vast amounts of data about proteins and are determining specific interactions among them, considerable further research is required to determine what each of these proteins does and how they all work together to form different living organisms. While the data collection itself is important as the networks (despite being incomplete and noisy) provide a holistic view of the functioning of the cell, it is clear that scientists need additional computational and experimental tools to analyze these biological networks in order to uncover cellular principles as well as protein functions and pathways. In this thesis, we introduce a new computational framework for analyzing biological networks in order to reveal and systematize cellular organization and functioning.

1.1 Biological networks

Proteins and their associated interaction networks are generally represented in a simple graph structure, with vertices representing proteins and edges representing the interactions among them. Each edge can be undirected or directed, depending on the

type of protein interaction it represents. Undirected edges are used to represent interactions between two proteins that interact physically; that participate in a synthetic lethal or epistatic relationship (i.e., in which mutations to the individual proteins do not cause loss of growth or fitness to the organism but do so in combination); or that are coexpressed (i.e., their transcripts are coregulated). Directed edges represent regulatory mechanisms and include transcription factor binding relationships (also called protein-DNA interactions) or where one protein phosphorylates or regulates another (review, [94]). Protein interactions can also represent functional associations, where two proteins are connected by an edge if they take part in the same biological process. Note that weights can be assigned to edges to represent the strength or confidence of an interaction.

1.2 Network analysis

Much work has been done to analyze biological networks computationally, providing us with hints of the inner workings of the cell. We describe those network analyses that are most closely related to our work; see [70,94] for excellent recent reviews with a broader scope. Initial topological analysis has suggested that biological networks are scale-free, that is they adhere to a power law distribution in which most proteins have only a few interactions but a few "hubs" have many interactions, which lends a robustness to the system and indicates that these networks have a structure different from that implied from an Erdos-Renyi random network model (which assumes that edges are independent and that each edge is equally likely) [3]. Densely connected subgraphs within interaction networks have been identified, and it has been shown that these "modules" [27] correspond to proteins taking part in the same physical complexes or the same biological processes [64, 77]. An examination of various interaction networks, including those with a mixture of interaction types, has shown that

certain combinations or patterns of interactions occur more frequently than expected by chance [46,50,62,73,92,93]. These oft-occurring topologies, called network motifs, are shown to be related to particular circuits in engineering, such as feed-forward loops [73].

Most interaction networks are imperfect, not only because they are incomplete, but because they provide only a static view of the interactome: they describe which proteins can interact, but most commonly fail to say anything about when the interaction occurs. Interactions may be determined *in vitro*, or via artificial constructs such as the two-hybrid system, or from cells exposed to a single condition or at a single time point. Network analysis most commonly groups these interactions together, while not considering spatial and temporal information. By incorporating gene expression data with the physical interaction network, dynamic properties of cellular networks may be uncovered. For example, an analysis of the hubs in the network has grouped them into two types: those that interact with all their partners in the same context (so that they function within a single cellular process) and those that interact with their neighbors at different times or in different locations (so that they link disparate biological processes) [24]. Furthermore, by integrating gene expression data with regulatory networks, it has been discovered that certain network motifs are more prevalent in specific cellular conditions [48].

The analyses presented so far have thoroughly ignored any features of the individual proteins in the network, and their focus has rested only on interactions. A recent line of work has attempted to incorporate the most basic of protein features, a protein's underlying amino acid sequence, in an attempt to identify conserved pathways within and between various organisms. These so-called network alignments involve comparing networks against each other by identifying homologs in the networks and then determining whether groups of these homologs interact in similar fashion in both networks [15,36,38,42]. Physical interaction networks have also been analyzed to re-

veal correlated sequence-signatures, or sequence motifs, in interacting pairs of proteins [78]; because the focus in this area [11, 20, 23, 63, 88] is on finding domain-domain interactions for the purpose of explaining interactions, only individual interactions in the network are considered.

1.3 Our contributions

Our work begins with the observation that while most existing network analysis has tended to focus solely on the topological features of interaction networks, there is much more information available to us about the proteins themselves, and this information can and should be used in the computational analysis. By incorporating protein features, such as the biological process in which a protein takes part or the domains it is predicted to contain, we expand the scope of the analysis. Patterns in the interaction network which are both feature and topology based more accurately describe the actual mechanisms by which processes in the cell work. Utilizing more abstract protein features than basic homology enables us to find a hierarchy of more abstract, recurring organizational units of increasing complexity in the interactome.

In this thesis, we first introduce the concept of *network schemas* to describe feature and topology based patterns in interation networks. In the remainder of Chapter 2, we develop algorithms for uncovering recurring, over-represented network schemas in physical interaction networks, and present a detailed analysis of schemas uncovered in the *S. cerevisiae* interactome. We demonstrate the functional importance of these schemas by showing that they correspond to functionally cohesive sets of proteins, are enriched in the frequency with which they have instances in the *H. sapiens* interactome, and are useful for predicting protein function. Next, in Chapter 3, we introduce fast algorithms and a powerful system for searching for all occurrences of a user-provided network schema in complex interaction networks. Our system gener-

alizes many previously studied types of interaction patterns, such as network motifs and domain-domain interactions, and allows a rich set of user queries. Finally, in Chapter 4, we consider future directions for research.

Chapter 2

Organization of Physical Interactomes as Uncovered by Network Schemas

2.1 Introduction

In this chapter, we introduce a framework for viewing networks in terms of organizational units consisting of specific, and potentially different, types of proteins that preferentially work together in various network topologies. Much previous work has focused on the topological properties of networks, identifying global topological and dynamic features [3, 24] and revealing a modular organization [27] with highly connected groups of proteins taking part in the same biological process or protein complex [64, 77]. Further analysis has shown that the wiring diagrams of biological networks are comprised of network motifs, or particular circuits, that occur more frequently than expected by chance [46, 48, 50, 62, 73, 92, 93].

Our goal is to explicitly incorporate known attributes of individual proteins into the analysis of biological networks. We conceptualize this with *network schemas*,

which are a general means for representing organizational patterns within interactomes where groups of proteins are described by arbitrary known characteristics along with the desired network topology of interactions among them (Figure 2.1A). A schema's matches, or instances, in an interactome are subgraphs of the interaction network that are made up of proteins having the specified characteristics which interact with one another as dictated by the schema's topology (Figure 2.1B). For example, a schema associated with signaling might be a linear path of kinases interacting in succession; its instances in *S. cerevisiae* include portions of the pheromone response and filamentous growth pathways. In graph-theoretic terms, a schema corresponds to a graph with labeled nodes and edges, and finding instances of a schema within an interactome corresponds to solving a subgraph isomorphism problem, which is known to be NP-complete.

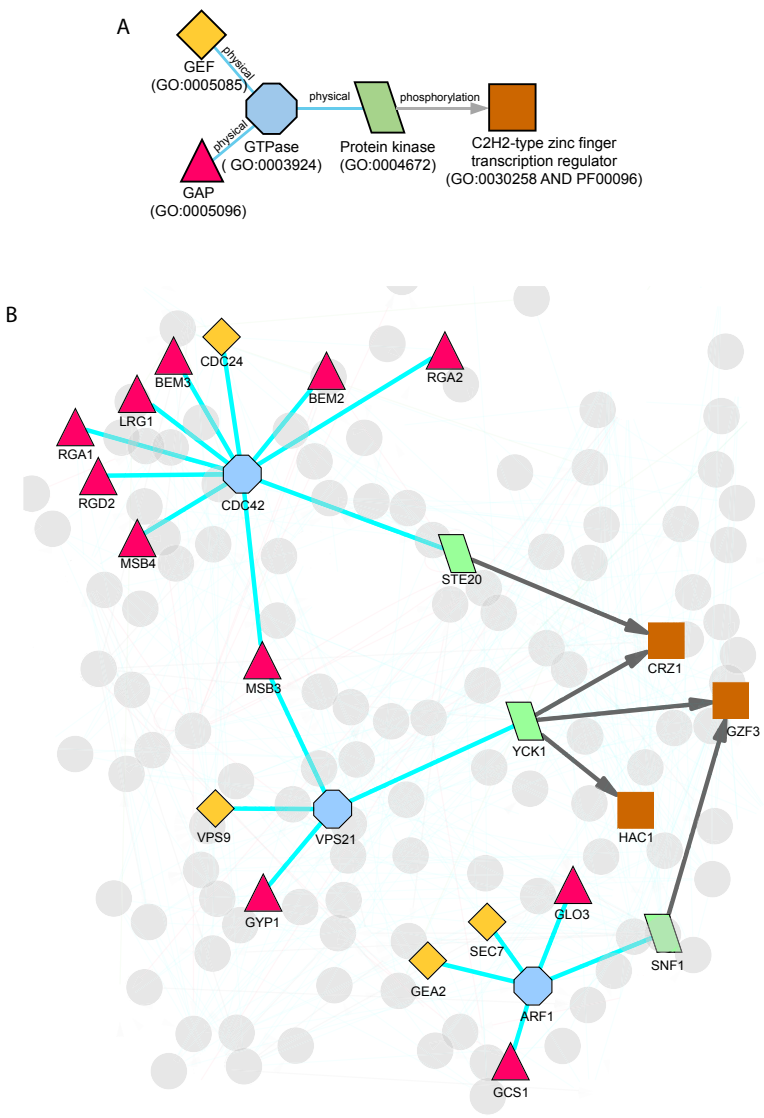

Figure 2.1: A sample schema and its instances in yeast. **(A)** An example of a schema. Each protein in the schema has a specific feature description and each edge has a type. In this case, the schema describes Ras GTPase signaling, where small G proteins from the Ras family are regulated by GTPase activating proteins (GAPs) and Guanine nucleotide exchange factors (GEFs), and in turn regulate effector kinases which may phosphorylate other proteins. **(B)** Instances of the schema in *S. cerevisiae*.

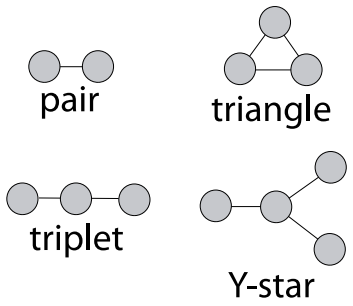

Figure 2.2: The network schema topologies that are considered in this study.

In this chapter, we present a system for analyzing physical protein-protein interaction networks and determining automatically which schemas are frequent and over-represented, and thus interesting enough to merit further analysis. Although any property can be used to annotate proteins in schemas, and different types of interactions may be specified, we focus on direct physical protein-protein interactions with proteins described via Pfam sequence motifs [4] and a set of GO molecular function terms [2]; such schemas with multiple instances in an interactome are likely to correspond to shared mechanisms that underlie a range of biological activities. Because we expect the largest number of schemas with multiple instances to be associated with small topologies, we begin to address these questions by considering four basic network topologies (Figure 2.2) varying from two interacting proteins (pair schemas) to higher-order schemas containing up to three interactions (triplet, triangle, and Y-star schemas); we choose these particular linear, cyclical, and branched topologies because they are the simplest patterns in physical interactomes that may intuitively be associated with signaling pathways, complexes, and switch-like patterns, respectively.

This work has three major contributions. First, we develop a computational procedure for automatically identifying *emergent* network schemas, or schemas that are both recurrent and over-represented in the interactome even when the frequencies of their lower-order subschemas are considered. Conditioning over-representation on

the distribution of a schema's lower-order constituents ensures that every emergent schema conveys novel information about interactome organization. We score a schema based upon its frequency in the interaction network and its expected frequency given the distribution of its constituent subschemas. The expected frequency is computed using a carefully designed graph randomization algorithm that preserves the distributions of the specific labeled subschemas. The false discovery rate (FDR) of the resulting scores is then evaluated using a variant of the permutation test. We note that in order to uncover emergent schemas, existing approaches for related problems could not be directly utilized; the specifics and scale of this problem required the development of novel computational techniques (see **Methods** for more details).

Second, in the first large-scale analysis of this type, we apply our procedure to the *S. cerevisiae* protein-protein interactome. In total, more than 140,000 Pfam network schemas that occur at least once in the *S. cerevisiae* interactome are considered. Of these, we identify 264 emergent Pfam network schemas with various annotations and topologies. We also uncover 138 emergent GO molecular function pair schemas. Analysis of emergent network schemas reveals a network organization where pair schemas are most diverse and where higher-order schemas reveal complex networks of primarily signaling and transport related activities. This suggests that the recurring units within interactomes are mostly pairwise, but that for some functions, higher-order recurring units are still prevalent. The hierarchical nature of emergent schemas can be visualized in a graph-theoretic manner which highlights that certain lower-order schemas occur frequently in higher-order emergent schemas (i.e., they are "hubs" in these networks), even though the frequencies of the lower-order schemas are controlled for in the computational procedure.

Third, we demonstrate that emergent network schemas correspond to biologically meaningful units. In particular, in a systematic analysis, we show that schema instances lead to protein subnetworks that share more specific biological process an-

notations than subnetworks having identical topologies but no constraints on the proteins making them up; this illustrates the additional benefit of incorporating protein annotations into traditional topology-based network analysis. Moreover, at the other extreme of the eukaryotic spectrum, we find that if we interrogate the *H. sapiens* interactome using the emergent schemas uncovered in *S. cerevisiae*, more than one-half of the schemas of each topology have instances there as well; this fraction is considerably lower when considering non-emergent *S. cerevisiae* schemas. Finally, we give a proof of concept through two uncharacterized protein families that network schemas can be used to functionally characterize protein families and individual proteins.

2.1.1 Relationship to previous work

Network schemas build upon earlier pioneering work in network analysis by enabling new types of analyses that were not possible with previous methods for identifying recurrent patterns in biological networks. By considering the specific roles of individual proteins, network schemas look beyond the purely topological features that are described by network motifs [46,48,50,62,73,90,92,93] to the tendency of certain types of proteins to work together, thereby shifting focus from the "syntax" of biological networks to their "semantics." While from a graph-theoretic point of view one may think of network schemas as a generalization of network motifs, considering protein attributes fundamentally changes what types of biological questions can (or cannot) be answered, and the much larger number of schemas changes the underlying computational issues as well. As compared to network alignments that uncover conserved interactions among homologous proteins in interactomes (e.g., [15,37,71,75]), network schemas utilize more abstract descriptions of proteins and are identified via a statistical model designed to find a hierarchy of interactome organizational units of increasing complexity. In contrast to approaches to uncover correlated sequence-signatures or

putative domain-domain or domain-peptide interactions via analysis of interactomes (e.g., [11, 18, 20, 23, 35, 53–55, 63, 78, 88]), network schemas incorporate higher-order topologies. Moreover, unlike the approaches that particularly focus on identifying domain-domain or domain-peptide interactions, schemas do not focus on the physical bases for protein interactions. Therefore, they represent more abstract organizational units, indicating what types of proteins work together and not which portions of the protein are responsible for the observed interactions. Further, it is important to note that combinations of pair schemas present in the interactome result in higher-order schemas that do not necessarily occur, and thus it is necessary to explicitly enumerate over these in order to uncover which exist in the interactome. Compared to a very recent approach for uncovering over-represented functional attributes in linear paths in regulatory networks [56], network schemas additionally consider cyclical and branched schema topologies, and their relationships to lower-order schemas.

2.2 Methods

2.2.1 Preliminaries

Protein annotations. We use Pfam [4] version 18.0 for motif annotations for all proteins. For *S. cerevisiae* proteins, we additionally consider a set of 134 general molecular function annotations from the Gene Ontology [2]. GO annotations for *S. cerevisiae* proteins are obtained for each sequence from SGD version 1.01 [29] utilizing all evidence codes. These GO terms have been selected by hand to maximize annotation coverage and minimize overlap with respect to GO.

Physical interaction network. We use *S. cerevisiae* and *H. sapiens* protein interaction data from BioGRID [80], release 2.0.20. Since we are interested in uncovering functional units consisting of proteins that work together in specific network topologies, we focus on direct physical interactions by utilizing interactions determined from

one of the following experimental systems: Biochemical activity, Co-crystal structure, Far western, FRET, Protein-peptide, Reconstituted complex, and Two-hybrid [29], excluding the IST 1 set of [34]. Additionally, interactions determined via Affinity capture-Western and Affinity capture-MS are used in the case where a bait protein identifies at most one prey. Proteins with ambiguous common names are not used. The physical interaction network is further filtered to remove: (1) interactions from a single experimental source for a protein if that source found over thirty interactions for this protein (2) any proteins with either less than one or more than fifty remaining interactions and (3) any proteins that do not have an annotation that appears at least twice in the remaining interaction network. After all filtering steps, the resulting Pfam-annotated *S. cerevisiae* network has 3,871 interactions among 2,073 proteins described by 472 Pfam terms, and the resulting *H. sapiens* network has 7,284 interactions among 4,062 proteins described by 669 Pfam terms. The same filtering process used with our set of GO molecular function terms on the *S. cerevisiae* interactome leaves 1,834 proteins with 3,542 interactions.

Terminology. A protein interaction network is represented as a labeled graph $G = (V_N, E_N)$, with a vertex $v \in V_N$ for each protein and an edge $(u, v) \in E_N$ between vertices whose corresponding proteins interact. Let $\mathcal{L}$ be the set of possible protein annotations (e.g., Pfam motifs). Each protein $v \in V_N$ is associated with a set of annotations $l(v)$, where $l(v) \subset \mathcal{L}$. A *network schema* is a graph $H = (V_S, E_S)$ where each vertex $v \in V_S$ is specified by a description $d_v \in \mathcal{L}$. An *instance* of a network schema H in an interaction network G is a subgraph (V_I, E_I) where $V_I \subset V_N$ and $E_I \subset E_N$ such that there is a one-to-one mapping $f : V_S \rightarrow V_I$ where for each $v \in V_S$, $d_v \in l(f(v))$ and there is an edge $(f(u), f(v)) \in E_I$ for each $(u, v) \in E_S$ (i.e., it is the match in the network for the schema). Note that two distinct instances of a schema may share proteins and/or interactions; however, any two instances must differ in at least one protein. Two instances of the same network schema are *independent* if they

are made up of non-overlapping proteins (i.e., the intersection of their vertex sets is empty). In the case of triplet and Y-star schemas, we allow instances that have additional interactions among the nodes in the interactome (i.e., the endpoints of the triplet or any pair of endpoints of the "spokes" of the Y-star may be connected with an edge). Note that network schemas can be naturally generalized to include other types of interactions and protein annotations (see Chapter 3).

2.2.2 Uncovering network schemas

The overall procedure for uncovering emergent network schemas of a given topology is as follows; the steps are described in more detail below. First, we count the number of instances of every schema that occurs in the interactome; though this corresponds to the NP-hard subgraph isomorphism problem, we find that in practice we are able to solve it readily (see Chapter 3). Second, for each schema that has at least two non-overlapping instances, we compute its average number of instances in randomized networks. Third, the schema is scored to favor schemas that both occur frequently and are over-represented compared to their average count in the randomized networks. Fourth, the significance of scores is determined using a false discovery rate that is calculated by repeating the first three steps of the process on randomized networks. Finally, the results are filtered in order to remove redundant schemas.

We developed an extensive algorithmic infrastructure as related techniques are not directly applicable. While there is substantial previous work in the data mining community for frequent (labeled or unlabeled) subgraph mining (e.g., see [10, 17, 30, 31, 33, 43, 44]), these approaches are focused on the algorithmic issues of enumerating (or eliminating) subgraphs in single or multiple networks, and not on assessing significance or relevance. Here, we are able to take a brute-force approach in enumerating subgraphs, and our methodology development instead is focused on identifying frequent and over-represented subgraphs. We further note that it is not possible simply

to apply the approach used for network motif finding [50] to uncover emergent network schemas as well. Specifically, in that approach the count of each network schema in the actual network would be compared to the count in randomized networks, and a p-value would be computed by considering what fraction of the randomized networks have a larger number of that network schema; however, this will identify as emergent schemas that occur rarely and are likely to be spurious but are made up of annotations that themselves occur rarely in the network, as these schemas are unlikely to be found in the randomized networks. A similar problem arises with using Z-scores, also reported in [50]. Our scoring and FDR procedure (described below) are designed to better handle the variation in annotation frequency and the large number of schemas of each topology that are considered. Finally, the task of building an ensemble of randomized networks that are constrained to have specified counts of certain labeled subgraphs has not, to the best of our knowledge, been addressed in the past.

Randomized networks for computing scores. For each schema s that recurs in an interactome (i.e., has at least two instances), we compute how often it occurs in randomized networks, which tells us whether the schema occurs more often than expected by chance. For each pair schema, we count how often it occurs in randomized networks that have been generated using the stub-rewiring approach of [50], which randomizes edges while maintaining the degree and labels of each node in the graph. Note that there is no known efficient method that generates graphs uniformly at random with specified degree and label distribution, so an approximation such as this is used. It is well known that the stub-rewiring procedure may result in networks where some nodes cannot achieve their desired degrees; however, we have found this to be rare in the networks studied here. For example, randomizing the *S. cerevisiae* network 100 times using stub-rewiring, we found that 98 of the random networks had all nodes reaching their original degrees, and 2 of the random networks had two nodes that are below their desired degree by 1. We note that while it is possible

to randomize the networks by shuffling annotations while keeping the topology fixed (e.g., as in [54]), annotations have different tendencies to be found in proteins of varying number of interactions, and we wish to maintain this relationship.

For each triplet and triangle schema, we count how often the schema occurs in networks randomized so as to preserve the distributions of the pairs making them up, and for each Y-star schema, we use the same approach, but consider randomized networks that preserve the distribution of triplets making up the Y-star schema (see below). In this manner, we are able to eliminate schemas that are over-represented only because they are comprised of lower-order schemas that are themselves over-represented; instead, we identify schemas that are over-represented even when considering the distribution of the lower-order schemas making them up. As with the stub-rewiring approach, the randomization methods for preserving pair and triplet distributions are approximate, as no efficient algorithms are known for these problems; however, as we show, they work well in practice.

We now describe the subgraph-preserving randomization methods in more detail. For each triplet schema where nodes labeled a and c interact with a central node labeled b, we generate randomized graphs that maintain the original number of interactions between proteins labeled a and proteins labeled b, and between proteins labeled b and proteins labeled c. Let these target interaction counts be denoted by t_{ab} and t_{bc}, and let s_{ab} and s_{bc} be the current count in the network we are generating. The counts of all other pairs of labelings are ignored. To generate the randomized graphs we repeatedly add edges between unconnected proteins, where the probability of adding a particular edge is proportional to how much closer it gets to the desired count of labelings, as measured by the squared L2 distance. That is, if node u is labeled with a and node v is labeled with b, an edge between them is added to the graph with probability proportional to $\max\{0, (t_{ab} - s_{ab})^2 - (t_{ab} - s_{ab} - e_{uvab})^2\}$, where e_{uvab} is the number of a-b labelings that are introduced by adding an edge between

u and v (in this case $e_{uvab} = 1$). Due to the fact that proteins often have multiple annotations, adding an edge may increase the count of more than one of the desired labeling pairs. In this case, the edge is added with probability proportional to the geometric mean of the individual pair labeling scores. We continue adding edges until the pairwise distributions are satisfied or no further edges can be added that can change the number of a-b or b-c labellings. As with the stub-rewiring approach, the degree of each protein is maintained, so that an edge is added only if the original degrees of both proteins have not yet been reached. Note that randomized networks are generated separately for each schema, and only edges changing the counts of constituent pair schemas are considered for addition into the network; that is, we only generate a small number of the edges (i.e., those that play a role in the corresponding lower-order schemas). This same process is used to generate randomized graphs for triangle schemas, except that a third pairwise count is also maintained (i.e., the $a - c$ count). The randomized graphs generated in this manner do an excellent job in achieving the desired distributions. For over 98% of all Pfam triplet schemas that have at least two independent occurrences in the original network (and 96% of triangles), the counts of all their constituent pairs are within one of their counts in the original graph for at least 90% of the randomized graphs.

The same overall scheme is adapted for randomizing networks in order to maintain triplet distributions. In particular, for each Y-star schema where a central node labeled with a interacts with nodes labeled with b, c, and d, randomized graphs are generated so as to maintain the number of paths where a protein annotated with b interacts with a protein annotated with a which in turn interacts with a protein annotated with c, the number of paths where a protein annotated with b interacts with a protein annotated with a which in turn interacts with a protein annotated with d, and the number of paths where a protein annotated with c interacts with a protein annotated with a which in turn interacts with a protein annotated with

d. We also consider the pairwise interactions in the Y-star; that is, the number of interactions between a protein labeled with a with a protein labeled with b, as well as the number of interactions with proteins labeled with c or d. An edge is added with probability proportional to the product of a pair term and a triplet term. As above, degree distributions are maintained and the pairwise (respectively triplet) term for each edge is the geometric mean over each pairwise (respectively triplet) labeling added depending on how much closer that edge gets one to the target count for that labeling. For each possible edge, the triplet term is initialized to be 1 until that edge can contribute to a triplet labeling. Once the pairwise term is 0 for all edges, only the triplet term is considered. This process is continued until the relevant triplet counts for the Y-star are satisfied or until no further edges can be added that can change these counts. At this point, if the randomized network has a triplet that has not reached its target count, we choose a protein that is annotated with the central label with probability proportional to its degree, and choose two proteins with the peripheral labels uniformly at random. New edges are added from the central protein to the two others, removing existing edges if necessary to satisfy the degree distributions. This process is repeated until all triplet target counts are met or exceeded. We find that for over 95% of Pfam Y-star schemas evaluated, the counts of all their constituent triplets are not less than one away from their counts in the original graph for at least 90% of the randomized graphs.

As mentioned, the randomization methods for preserving degree distribution, and pair and/or triplet subschema distributions are approximate and do not come with theoretical guarantees. In order to show that the described randomization procedure produces networks that are sufficiently different from one another (i.e., sample a wide range of possible networks), we take the five top-scoring schemas of triplet, triangle and Y-star topologies and generate 1000 subschema preserving randomized networks for each of them. We then calculate the overlap between each pair of randomized net-

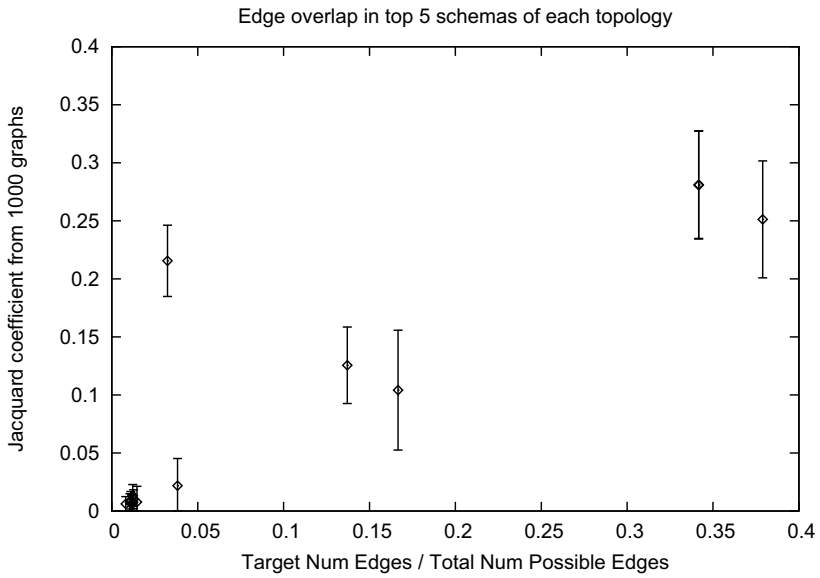

Figure 2.3: Variation in 1000 subschema preserving randomized networks for each of the five top-scoring triplet, triangle, and Y-star schemas. Each point in the graph plots the average Jacquard coefficient between pairs of networks randomized for the same schema, with error bars showing plus and minus one standard deviation, as a function of the total target number of edges desired between proteins of particular annotations divided by the total number of possible edges having those annotations. The overlap between randomized networks also appears to depend on the degree distribution of the proteins with the relevant labels (not depicted here).

works for each schema as the Jacquard coefficient over the edges present in each of the networks. The low average pairwise overlaps (Figure 2.3) indicate that the randomization procedure is sampling broadly from the set of possible networks. Moreover, we observe that for a given schema the average overlap between subschema preserving networks seems to depend on the number of the target edges desired, the total number of possible edges of the appropriate labels possible, and the degree distribution of the nodes annotated with the labels of interest.

Scoring schemas. For each schema s, let $count_s$ be the number of times it occurs and avg_s be the average number of times it occurs in randomized networks. The score for schema s is given by

$$(count_s + 1) \log \left(\frac{count_s + 1}{avg_s + 1} \right).$$

The addition of the pseudocount of 1 downweighs the contribution of very rare schemas that could otherwise obtain high scores simply due to very small (or zero) average counts in the randomized graphs. The scoring function takes into account both a schema's frequency and its over-representation in the real graph compared to the randomized one. While other scoring functions may be utilized, we note that due to the variation in how frequent various annotations are, $count_s$ by itself is not an ideal choice as it favors schemas comprised of frequent annotations.

For each schema, 100 randomized networks are generated, and the average number of times that each schema occurs in these networks is computed. Overall results did not change appreciably when considering more randomizations in this step and keeping the rest of the framework the same (data not shown), suggesting that 100 randomizations are adequate for our purposes. Due to computational concerns, and since we are only interested in independent recurring schemas, scores are computed only for the 419 pair, 842 triplet, 31 triangle, and 999 Y-star schemas that occur independently at least twice in the interactome.

Significance model. For each putative recurring schema found in the real network, we obtain a score reflecting its frequency and over-representation compared to the randomized graphs. In order to evaluate the significance of these scores, for each schema topology, we repeat this procedure with multiple *iteration* graphs created by the stub-rewiring algorithm of [50]. Since all associations in these randomized networks occur by chance, we can use them to calculate the FDR for each score, or

the fraction of schemas with score $\geq s$ that arise from chance alone. For n iteration graphs, it can be computed as

$$\frac{\frac{1}{n}\sum_{\text{iteration graph i}} \#\text{ putative schemas in graph i with score } \geq \text{s}}{\#\text{putative schemas in the real graph with score } \geq \text{s}}.$$

Here, $n = 50$ iteration graphs are used. In order to correct for differences in the clustering coefficient between real and randomized graphs, the FDR of triangle schemas is further corrected by multiplying by the ratio of the number of triangles in the actual network to the average number found in randomized graphs. Note that the false discovery rate corrects for multiple hypothesis testing. We use an FDR of ≤ 0.05 as the significance cutoff to identify emergent schemas. Note that other FDR values can be used as a cutoff to identify emergent schemas; we choose the 0.05 level as it is a commonly-used one that appears reasonable in this application.

Filtering schemas. Once schemas over-represented at FDR ≤ 0.05 are identified, we eliminate any schema for which at least 15% of the randomizations have a labeled subgraph whose count is more than one below its count in the original network. Additionally, the instances for these schemas are obtained and we eliminate those schemas whose instances are a subset of the instances of another schema from the same topology. The remaining schemas are our uncovered emergent schemas.

Network alterations. In order to check whether the schemas identified as emergent are robust to changes in the network, we recompute FDRs on the yeast network altered in the following way. First, we remove a percent x of the interactions, where each such interaction is chosen uniformly at random. We then add an equal number of interactions, where the two proteins to be connected are again chosen uniformly at random. We consider altered networks with $x = 2.5\%, 5.0\%, 7.5\%$ (i.e., resulting in networks differing from the original network by up to 5%, 10%, and 15% respectively), and generate five altered networks for each of these values. For each perturbed

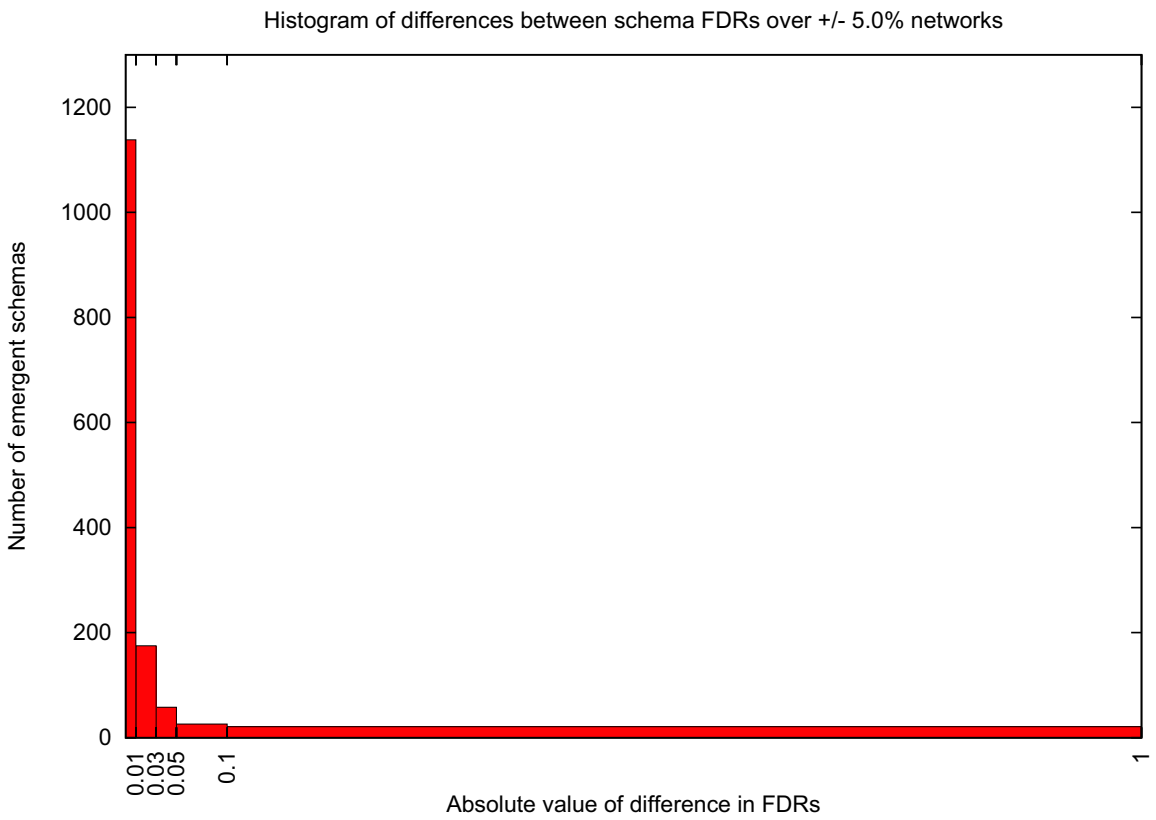

Figure 2.4: A histogram of the absolute differences between the FDRs of emergent schemas in the original network and a network altered by removing and then adding 5.0% of the edges. Results in each histogram are aggregated over five altered networks, and the heights of the bars give the number of schemas falling into the five bins corresponding to changes in FDR < 0.1, 0.03, 0.05, 0.1 and 1.0.

network, the absolute value of the difference in FDRs over all schemas identified as emergent in the original network is computed. Figure 2.4 gives a histogram of these values over the 5.0% perturbed networks and shows that the FDRs for most emergent schemas vary very little, with a few outliers. For the networks altered by removing 2.5% and adding 2.5% of the interactions, the median absolute change in FDR over emergent schemas varies from 0.0017 to 0.0036 in the five perturbed networks; these numbers are 0.0018 to 0.0033 when adding and removing 5%, and 0.0019 to 0.0043 when adding and removing 7.5%.

Computing requirements. The described schema discovery process is run on a Dell Linux Cluster with 3.2 GHz Xeon and 3.0 GHz Woodcrest processors; 51 total nodes are used (one for the FDR computation of the original network and one for each of the iteration graphs). The entire process for uncovering schemas of the four topologies considered typically takes 12 total hours in a shared user environment.

2.2.3 Evaluating functional coherence

For each topology, we compile the set of instances of all Pfam emergent schemas. Duplicate instances are removed; for the "background" set, we enumerate all subgraphs of that topology in the same filtered interaction network that is used to search for the Pfam schemas. To avoid any bias that might arise from Pfam annotations, only proteins having at least one Pfam annotation are considered when building the background sets of subgraphs. Furthermore, we require all proteins in each schema instance and each background subgraph to have non-trivial GO biological process annotations; in the case of the Y-star topology, this requirement is relaxed to permit the central node to be unannotated. For each such subgraph, we determine the least common ancestor (LCA) of the annotations of the proteins in the GO biological process graph; if there are multiple LCAs, we select the one that annotates the smallest number of proteins in *S. cerevisiae*. Note that if the proteins are not known to be functionally related, the LCA of their annotations would be the trivial annotation of *biological_process*. The "specificity" of this LCA is calculated as the probability p of a schema-sized set of proteins having that annotation, using the hypergeometric distribution. Finally, for a given value of p, for both the emergent schema instances and the background set of subgraphs, we can measure the functional coherence of each as the fraction of subgraphs whose constituent proteins have annotations whose LCA specificity is at most p.

2.3 Results

2.3.1 Emergent network schemas in the *S. cerevisiae* interactome

Each pair schema is scored by considering its number of occurrences in the *S. cerevisiae* interactome against its average number of occurrences in degree-preserving random networks [49, 50, 73]. Each triplet, triangle, and Y-star schema is scored similarly, except that its average number of occurrences is computed in networks randomized so as to maintain the distribution of its constituent pairs (for triplet and triangle schemas) or its constituent triplets (for Y-star schemas). Using a false discovery rate (FDR) of ≤ 0.05, we identify 151 pair, 55 triplet, 26 triangle, and 32 Y-star Pfam emergent schemas in the *S. cerevisiae* network comprised of direct physical interactions. The emergent schemas are a small fraction of the total number of schemas occuring in the interactome. In total, 2838 pair, 24662 triplet, 999 triangle and 114650 Y-star Pfam schemas occur at least once in the *S. cerevisiae* interactome. Of these, 419 pair, 842 triplet, 31 triangle, and 999 Y-star schemas are recurring in that they have at least two non-overlapping instances (i.e., that do not contain a protein in common).

The emergent pair schemas are depicted in a network in Figure 2.5A. Pair schemas represent two proteins working together (as a dimer or as part of a complex), or one protein (de)activating another. The uncovered emergent schemas represent a wide variety of functions including signaling (e.g., schemas involving *Pkinase* or *Ras* motifs), transport (e.g., schemas involving the amino acid permease motif *AA_permease*), intracellular trafficking (e.g., *synaptobrevin* schemas), RNA processes (e.g., *RRM_1* schemas) and ubiquitination (e.g., ubiquitin-conjugating enzyme motif *UQ_con* schemas). While some of the pair schemas may correspond to actual domain-domain interactions, the schema formulation by itself does not make any claims about the interaction

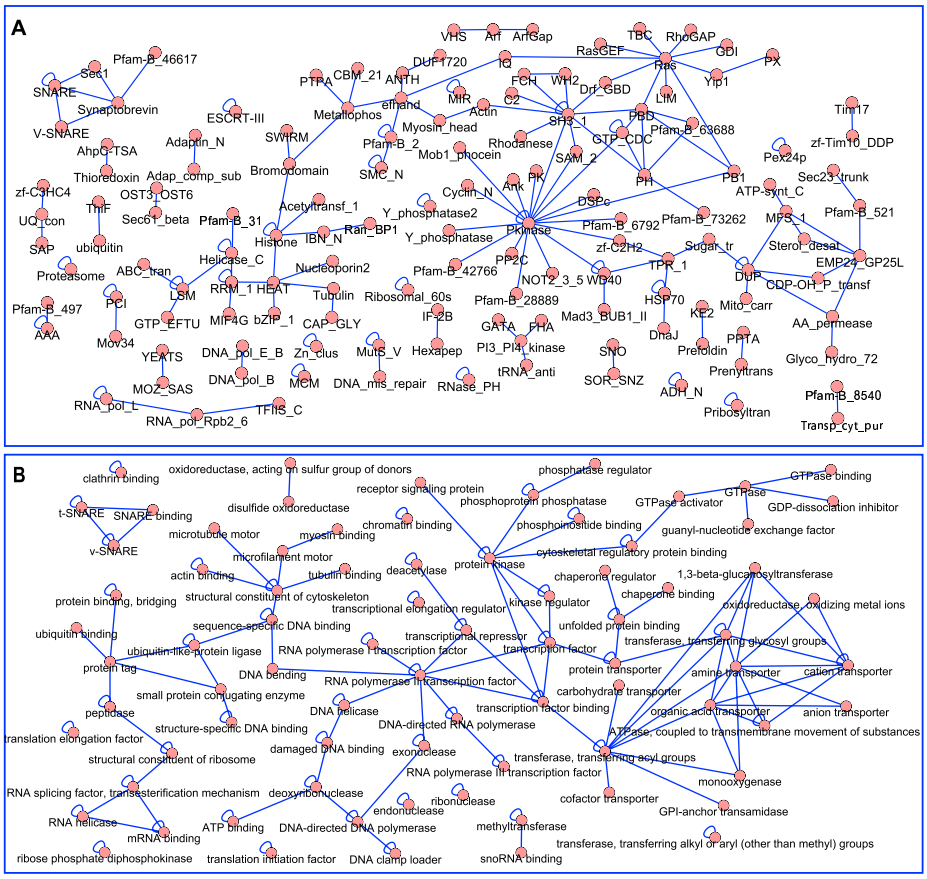

Figure 2.5: Emergent pair schemas uncovered in the *S. cerevisiae* interactome. A pair of vertices connected by an edge corresponds to a pair schema. **(A)** Pfam emergent pair schemas, where each vertex is labeled with a Pfam motif. **(B)** Gene Ontology molecular function emergent pair schemas, where each vertex is labeled with a GO molecular function term, with the word "activity" dropped from term names.

interface. In particular, some of the underlying physical interactions may instead consist of domains interacting with peptides or disordered regions [41]. This is clear, for example, when looking at the diverse set of pair schemas involving the *SH3* domain which is known to typically bind proline-rich peptides [84]. Nevertheless, similar to earlier findings for domain-domain interactions [35], we find that emergent Pfam pair schemas are enriched in homotypic annotations as compared to all Pfam pair schemas in the interactome (18.5% vs. 5.8%).

We also uncover *S. cerevisiae* emergent pair schemas using a hand-chosen set of GO molecular function annotations (Figure 2.5B). As with the Pfam schemas, the GO

pair schemas represent many types of functions including transport, signaling, DNA and RNA processing, ubiquitination, protein folding, and cytoskeleton organization. The GO molecular function schemas can sometimes allow generalizations of the Pfam schemas that move beyond sequence similarity, as proteins annotated with the same GO molecular function term need not be homologous to each other. For example, the Pfam pair schema consisting of a protein with the *Pkinase* motif interacting with a protein with the cyclin N-terminal motif *Cyclin_N* is subsumed by the GO schema consisting of a protein with kinase activity interacting with a protein with kinase regulator activity. Instances of this GO schema in the *S. cerevisiae* interactome include cyclins which lack the *Cyclin_N* Pfam motif, other cyclin-like proteins, and different kinase regulators altogether, such as activating subunits of kinase complexes, adaptors, and scaffold proteins. As another example, the Pfam pair schema consisting of the *Pkinase* motif interacting with the zinc finger motif *zf-C2H2* has a correspondence in a GO schema consisting of a protein with kinase activity interacting with a protein with transcription factor activity; instances of the latter schema in the *S. cerevisiae* interactome include transcription factors of the zinc finger, MADS, and basic helix-loop-helix families.

Higher-order emergent *S. cerevisiae* network schemas are given in Figures 2.6 and 2.7. For the purpose of visualization, they are represented as networks where vertices correspond to lower-order schemas. That is, for each higher-order schema, there is a vertex for each of its corresponding lower-order schemas, along with edges between these vertices; triplets and triangles are depicted with respect to lower-order pair schemas whereas Y-stars are depicted with respect to lower-order triplet schemas (see Figures 2.6A, 2.6C, and 2.7A for explanation). Edges in these networks thereby indicate that the two corresponding lower-order schemas are found together as parts of a emergent higher-order schema.

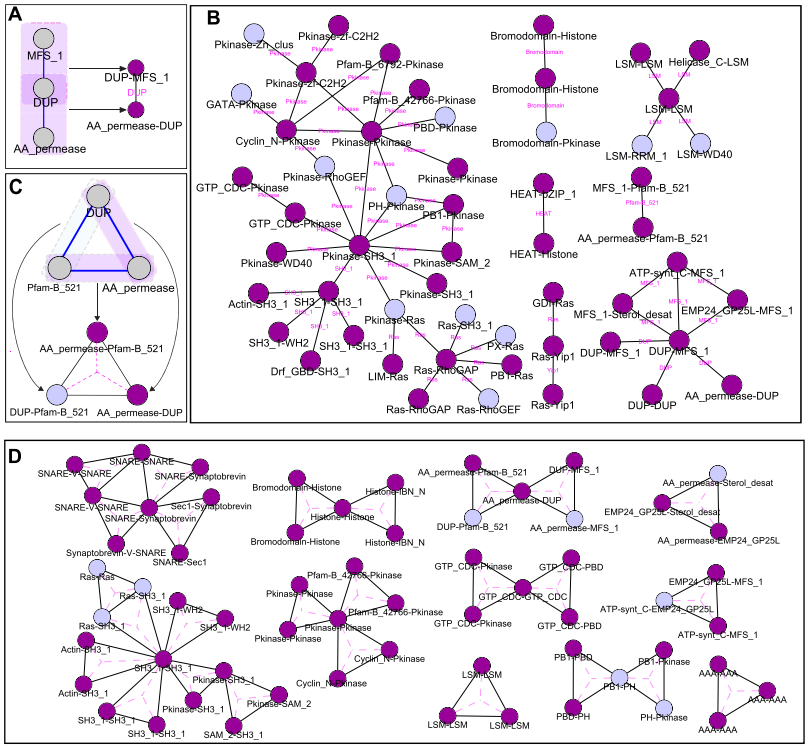

Figure 2.6: Emergent triplet and triangle schemas uncovered in the *S. cerevisiae* interactome, represented in a graph where vertices correspond to pair schemas. Pair schemas that are themselves emergent (Figure 2.5) are displayed as darker vertices. **(A)** An illustration of the subgraph representation for triplet schemas. The triplet *MFS_1-DUP-AA_permease* (on the left) is mapped to two pair vertices, corresponding to the lower-order pair schemas making it up, connected by an edge. The edge is labeled in pink with the central motif of the triplet (DUP). **(B)** Pfam emergent triplet schemas. **(C)** An illustration of the triangle schema *DUP-AA_permease-Pfam-B_521*. The triangle *DUP-AA_permease-Pfam-B_521* is mapped to three pair vertices, corresponding to the lower-order pair schemas making it up, connected by edges; that is, it is represented as a triangle in the graph whose vertices represent pair schemas. The *DUP-Pfam-B_521* pair, colored pale in the pair-vertex graph, is not an emergent pair schema, whereas the other two pairs in the triangle, colored dark in the pair-vertex graph, are. **(D)** Pfam emergent triangle schemas.

The uncovered emergent triplet schemas (Figure 2.6B) include several relating to signaling (e.g., *Pkinase* and *Ras* schemas) and transport (the connected components with the *MFS_1* motif). The signaling schemas include kinase cascades (e.g., *Pkinase-Pkinase-Pkinase*), regulation of Ras signaling (e.g., *RhoGAP-Ras-RhoGEF*), those connecting Ras and kinase signaling (e.g., *RhoGAP-Ras-Pkinase*), and those relating to specific structural domains involved in signaling [59] (e.g., *SH3-Pkinase-*

WD40 and *SH3-Pkinase-PH*). Note that there are many possible schemas associated with signaling (e.g., consider the set of schemas annotated with all domains known to be associated with signaling [59]), and our schema analysis identifies only a small subset of these as emergent. There are numerous emergent triplet schemas involving the major facilitator superfamily (*MFS_1*), one of the two largest families of membrane transporters [57]. Triplet *MFS_1* schemas include those involving other transport proteins, such as membrane proteins involved in transport of amino acids (i.e., containing the *AA_permease* motif) and proteins involved in ER to Golgi transport (e.g., containing the *EMP24* motif). Whereas the pervasiveness of kinases within conserved portions of the interactome has been observed earlier [37], the prevalence of such transport related subnetworks has been previously underappreciated.

Many of the triangle schemas (Figure 2.6D) correspond to known complexes. There are several triangle schemas, making up a connected component, corresponding to the SNARE vesicle-fusion machinery. The triangle schema made up of *LSM* motifs corresponds to Sm and LSM complexes, and is associated with the spliceosome as well as other RNA processing [28]. The triangle schema made up of *AAA* motifs corresponds to replication factor C complex and the 19S particle of the 26S proteosome. There are numerous triangle schemas associated with signaling as well; these may correspond, for example, to complexes or phosphorylation by kinase complexes. For example, the *Cyclin_N-Pkinase-Pkinase* triangle schema contains instances where a cyclin associates with a cyclin-dependent kinase, and this complex either phosphorylates or is phosphorylated by another kinase.

The emergent Y-star schemas (Figure 2.7B) refine the functional landscape of the triplet schemas, with one relating to transport and several relating to *Ras* and kinase signaling pathways. The Y-star schemas showcase the complex, nonlinear regulatory patterns evident in biological pathways. For example, some of the Y-star *Pkinase* schemas relate to the role of phosphorylation in combinatorial regulation of tran-

scription factors (e.g., those including multiple transcription factor motifs, such as *zf-C2H2* and *GATA* interacting with the same kinase), whereas others correspond to kinase cascades that additionally incorporate regulation via cyclins (e.g., schemas including *Cyclin_N*). Additionally, several Y-star schemas represent a dynamic "switch-like" pattern in which the peripheral proteins are active in different contexts. This is evident in some schemas where the peripheral proteins belong to the same family, and utilize the same structural interface on the central protein. For example, several of the Y-star Ras schemas consist of a central Ras protein interacting with several regulatory GTPase activating proteins (corresponding to *RhoGAP*, *TBC* or some *LIM* containing proteins). Such schemas show that certain types of "mutually exclusive" interactions [40] recur together in the interactome.

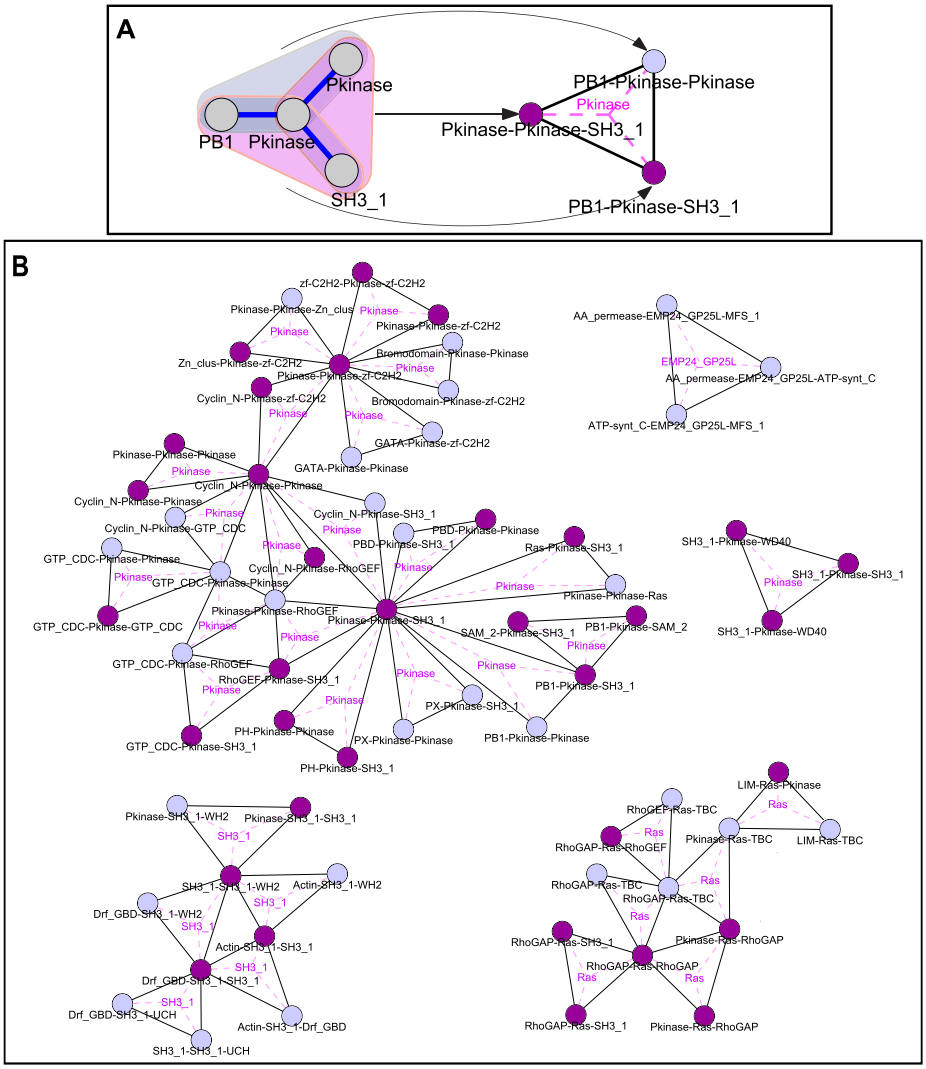

Figure 2.7: Emergent Y-star schemas uncovered in the *S. cerevisiae* interactome, represented as triangles in a graph where each vertex corresponds to a triplet schema. Triplet schemas that are themselves emergent (Figure 2.6) are displayed as darker vertices. **(A)** An illustration of the triplet subgraph representation of a Y-star schema. The Y-star (on the left) is mapped to three vertices corresponding to its lower-order triplet schemas, along with edges among them; that is, it is represented as a triangle in the graph whose vertices represent triplet schemas. The triplet subschemas of the Y-star are highlighted. The subschemas that are emergent triplets are highlighted in purple and represented as darker vertices. For ease of visualization, the central node of the Y-star is labeled in pink inside the triangle and connected to the vertices by dashed lines. **(B)** Pfam emergent Y-star schemas.

2.3.2 Schemas give insight into organizational principles of interactomes

While each emergent network schema represents a specific way in which proteins can work together, their relationships to one another, and in particular of higher-order schemas to lower-order ones, lead to some general observations about network organization.

The first observation is the striking drop in the number and diversity of emergent schemas with increased complexity, especially between pair and higher-order schemas (there are 139, 39, 29 and 30 distinct Pfam motifs involved in pair, triplet, triangle and Y-star emergent schemas respectively). Whereas 36% of the recurring pair schemas in *S. cerevisiae* are found to be emergent, only 6% of recurring triplet and 3% of recurring Y-star schemas are. (Note that triangle schemas are something of a special case because the cyclical structure is very constrained and recurring units are unlikely to be found at random.) This suggests that the semantic units within interactomes are primarily at the pair level, and that most repeated patterns of higher order can be viewed as rearrangements of the pairs that can be explained simply by randomness. At the same time, there are a considerable number of higher-order schemas (i.e., those identified as emergent) that cannot be explained by lower-order ones.

These higher-order emergent schemas are not just combinations of the lower-order emergent pair schemas. For example, the emergent pair schema network (Figure 2.5) contains 712 triplets, of which 571 occur even once in the *S. cerevisiae* interactome. Of these, only 37 are emergent. Thus, the majority of possible triplets resulting from emergent pair schemas are not emergent, and triplet schemas thereby allow us to uncover which sets of proteins comprising pair schemas work together in the network. On the other hand, 18 emergent triplet schemas are not present in the emergent pair schema network. For example, the *RhoGAP-Ras-Pkinase* emergent triplet schema consists of the *Ras-Pkinase* pair which is not found to be emergent. Though this

pair occurs numerous times in the network, given the frequency of *Ras* and *Pkinase* Pfam motifs, it does not appear at the FDR ≤ 0.05 level; this also demonstates that, as intended, our procedure for uncovering schemas corrects for the frequency of the motifs.

Large fractions of the distinct lower-order schemas making up the higher-order emergent schemas are themselves emergent (73% and 80% of the pair schemas comprising triplet and triangle schemas, respectively, and 51% of the triplet schemas making up Y-star schemas). The use of subgraph-preserving randomizations in our procedure confirms that this observation is not due solely to the abundance of the lower-order structures, but is a more general feature of schema organization. This result has a topological counterpart, as it has been found that four-protein network motifs tend to be combinations of three-protein ones [92].

Several emergent schemas from each topology share particular lower-order schemas. These lower-order schemas that are found in numerous higher-order schemas correspond to hubs in Figures 2.5, 2.6, and 2.7. We observe that the nodes with largest degree in the *S. cerevisiae* Pfam pair graph (Figure 2.5A) are *Pkinase*, *SH3_1*, and *Ras*. These domains comprise hubs at different levels of schema complexity. For example, the pairs that are hubs in the triplet graph (Figure 2.6B) are *Pkinase-SH3_1*, *Ras-RhoGAP*, and *Pkinase-Pkinase*. It is instructive to compare these families to the list of the 10 most frequent Pfam motifs and the 10 Pfam motifs involved in the highest number of interactions in the studied network. As expected, because of our scoring procedure which considers the frequency of annotations in the network, while some of the "hub" motifs are frequent in the interactome or common in interactions (e.g., *Pkinase* and *SH3*), many are not (e.g., *RhoGAP*); additionally, there are many Pfam motifs that occur frequently in the network but are not prevalent in these schemas (e.g., *Helicase_C*).

2.3.3 Schemas recapitulate known biology: the Ras superfamily

As an illustrative example showing that automatically uncovered emergent schemas can show excellent correspondence to well-understood organizational and functional units, we detail our findings on *S. cerevisiae* emergent Pfam schemas involving the Ras superfamily. There are ten *Ras* pair schemas (Figure 2.5A). The *Ras-RhoGAP*, *Ras-RasGEF*, and *Ras-TBC* schemas correspond to the basic regulatory interactions of Ras proteins. The *Ras-GDI* pair reflects the additional regulatory mechanism of the Rab subfamiliy of Ras proteins by the guanyl dissociation inhibitors (GDIs). The *Yip1* family of proteins in turn may act as GDI displacement factors [76] for a group of Ras-like proteins associated with Golgi membranes and/or act as membrane recruiters of these proteins [91]. Two Ras pair schemas involve Ras-binding motifs—the diaphanous GTPase-binding motif *DRF_GBD* found in Rho effectors and the P21-Rho-binding motif (*PBD*). Other Ras pair schemas contain motifs that reflect the biological role of Ras families, such as the *IQ* calmodulin-binding motif and the *PB1* domain associated with signaling. Finally, *LIM* is a general structural domain, but is found in several GAP proteins. The higher-order Ras emergent schemas (Figures 2.6 and 2.7) include several that reflect their diverse regulatory mechanisms. For example, there is a *Pkinase-Ras-RhoGAP* triplet, where the *RhoGAP* regulates the *Ras* which in turn regulates the kinase, and a *RhoGEF-Ras-RhoGAP* triplet, where both the *RhoGEF* and *RhoGAP* regulate *Ras*.

2.3.4 Schemas uncover functionally coherent portions of the interactome

To validate in a systematic manner that emergent schemas correspond to functional units and may be helpful towards uncovering network modularity, we determine

whether individual instances of emergent schemas have enriched functional coherence beyond that suggested by guilt-by-association and subgraph topology. As described in **Methods**, for each topology we determine the specificity, estimated using the hypergeometric distribution, of the most descriptive biological process annotation shared by the proteins in an instance of an emergent schema. For the background set, we enumerate all subgraphs of a given topology in the interaction network, with the restriction that only proteins having at least one Pfam annotation are considered (to avoid bias arising from Pfam annotated proteins). We find that 77% of the instances of the emergent pair schemas share a biological process at the $p \leq 0.01$ level, as opposed to 53% for the background set. These numbers are 60% vs. 35% for triplet schemas, 87% vs. 69% for triangle schemas, and 58% vs. 21% for Y-star schemas. This enrichment is observed over the entire range of p-values (see Figure 2.8). Functional enrichment is likely due in part to the enrichment of true interactions in emergent schema instances; indeed, interactions from small-scale experiments (< 50 interactions uncovered total) are enriched in the emergent pair Pfam schemas instances as compared to the entire interactome.

2.3.5 Enriched number of *S. cerevisiae* emergent network schemas with instances in *H. sapiens*

In order to determine whether emergent *S. cerevisiae* schemas tend to be found in other organisms, we have used each schema to interrogate the full (i.e., unfiltered) *H. sapiens* physical interaction network in BioGRID [7] and obtain its instances. We limit this analysis to schemas comprised of Pfam annotations that occur in both *S. cerevisiae* and *H. sapiens*. We find that 76% of these *S. cerevisiae* Pfam emergent pair schemas have at least one instance in the *H. sapiens* network. For comparison, if we consider pair schemas with instances in *S. cerevisiae* with FDR > 0.05, only 38% have instances in *H. sapiens*. The fraction with instances in *H. sapiens* is 75% for

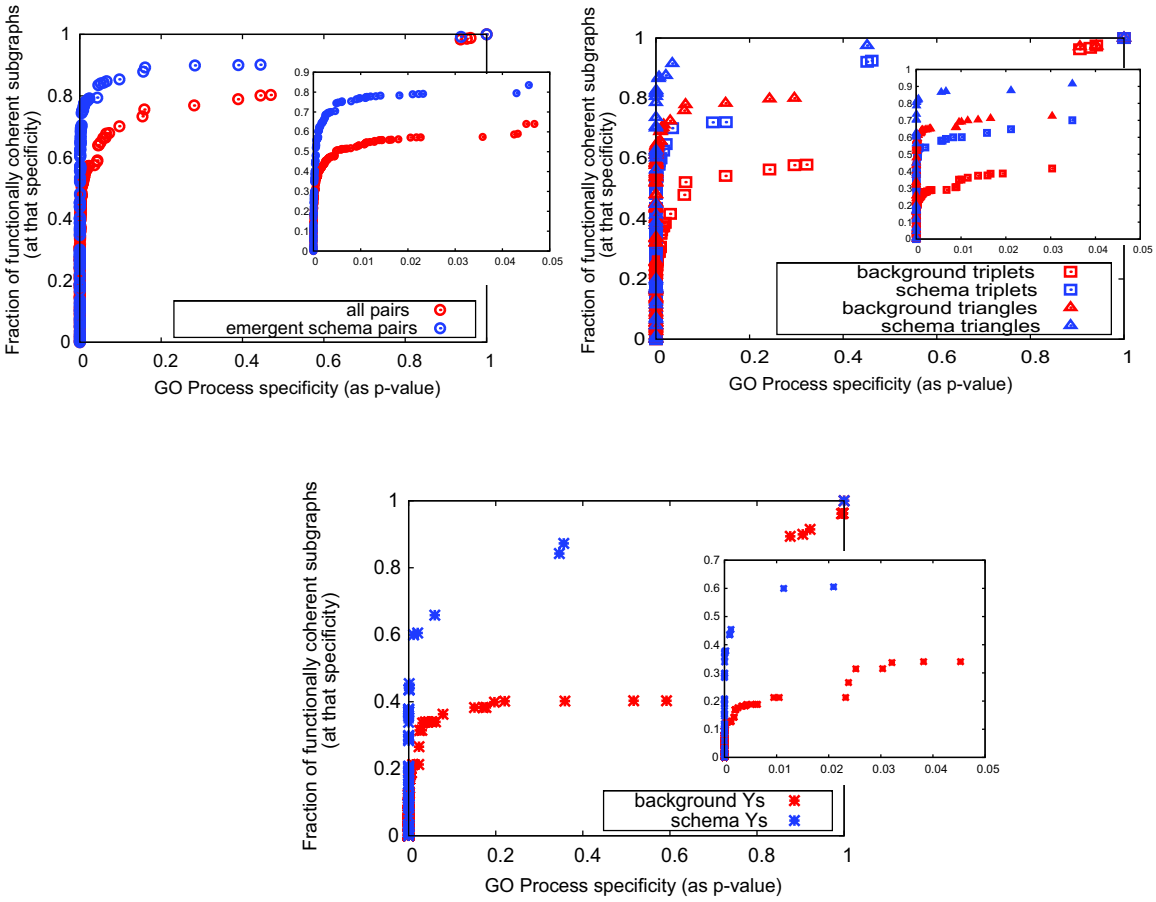

Figure 2.8: Functional coherence of emergent schema instances compared with arbitrary subgraphs of the same topology. Each panel compares emergent schemas (shown in blue) with a background set of schemas (shown in red) with respect to biological process coherence. As a function of a particular p-value p, we plot the fraction of schema instances that share a biological process term that has p-value less than or equal to p, as judged by the hypergeometric (see text). For all topologies (pairs, first panel; triplets and triangles, second panel; Y-stars, third panel) and over the entire range of p-values, the emergent schemas have a higher fraction of instances with shared biological process than background schemas of the same topology.

emergent triplet schemas, 61% for emergent triangle schemas, and 55% for emergent Y-star schemas; the instance percentages for schemas not found to be over-represented are 17%, 15%, and 8% respectively. Thus, emergent schemas have instances in *H. sapiens* two to seven times more frequently than schemas of the same topology that are not found to be over-represented, giving further evidence that these schemas correspond to recurring units within interactomes.

2.3.6 Network schemas in the *H. sapiens* interactome

To compare the types of schemas that are emergent across distant genomes, we uncover pair schemas in the *H. sapiens* interactome (Figures 2.9 and 2.10). We identify 29 pair schemas that are emergent schemas in both the *S. cerevisiae* and *H. sapiens* networks, as well as several that are emergent schemas only in *H. sapiens* but have instances in *S. cerevisiae* (Figure 2.9). As expected, these schemas represent some of the most basic processes that occur within the cell: DNA packaging, cytoskeleton organization, signaling, vesicle fusion, and so on.

The *H. sapiens* emergent pair schemas that are not found in *S. cerevisiae* (Figure 2.10) contain many schemas related to processes specific to higher organisms. These include, for example, schemas involving the extracellular matrix (e.g., *Collagen* and *Fibrinogen_C* schemas) and intercellular signaling (e.g., *Hormone_recep* schemas), among others. Many of these types of schemas consist of Pfam motifs that are not found in *S. cerevisiae* (e.g., the *Death* domain, found in proteins associated with apoptosis). The *H. sapiens*-specific emergent pair schemas also include some where both motifs are also found in *S. cerevisiae*; some of these schemas correspond to expansions of protein families and their interactions in *H. sapiens*. These include, for example, several emergent schemas involving motifs that are associated with phosphotyrosine signaling (e.g. *SH_2* and *Y_phosphatase* schemas); though these motifs are found in *S. cerevisiae*, they are rare. Additionally, the *H. sapiens* emergent pair schemas reveal how newer motifs, found only in *H. sapiens*, are integrated into networks containing older motifs, found in both organisms. For example, the tyrosine kinase *Pkinase_tyr* motif, found in *H. sapiens* but not *S. cerevisiae*, is involved in emergent pair schemas with signaling domains such as *SH3_1* and *PH* that are found in both organisms.

The *H. sapiens* and *S. cerevisiae* schemas taken together help fill in some of the data missing from the current state of interactomes, as combining the emergent schemas from the two interactomes gives a more complete view for some processes.

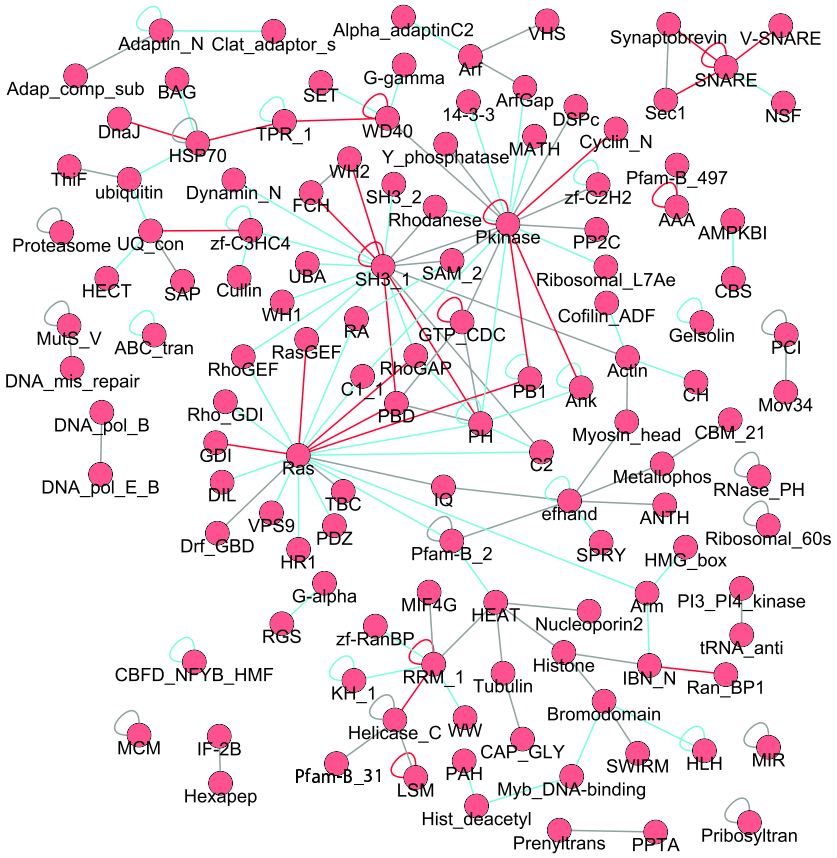

Figure 2.9: Pfam pair schemas that are found in both *H. sapiens* and *S. cerevisiae*. Schemas that are emergent in both organisms are displayed with red edges. Schemas that are emergent only in *H. sapiens* but that have instances in *S. cerevisiae* are shown with light blue edges. Schemas that are emergent only in *S. cerevisiae* but that have instances in *H. sapiens* are indicated with grey edges.

For example, several schemas relating to ubiquitination consist of pairs that are found to be emergent in only one organism but which have instances in the other; this is most likely due to missing interactions in one of the interactomes. The *S. cerevisiae* emergent schemas cover two parts of the ubiquitination pathway: they include an interaction between the *ubiquitin* family and the *ThiF* family of ubiquitin-activating enzymes, which catalyze the first step of the pathway, and an interaction between the *UQ_con* family of ubiquitin-conjugating enzymes and the *zf-C3HC4* (RING finger) family of ubiquitin ligases, which catalyze the second and third steps of the pathway,

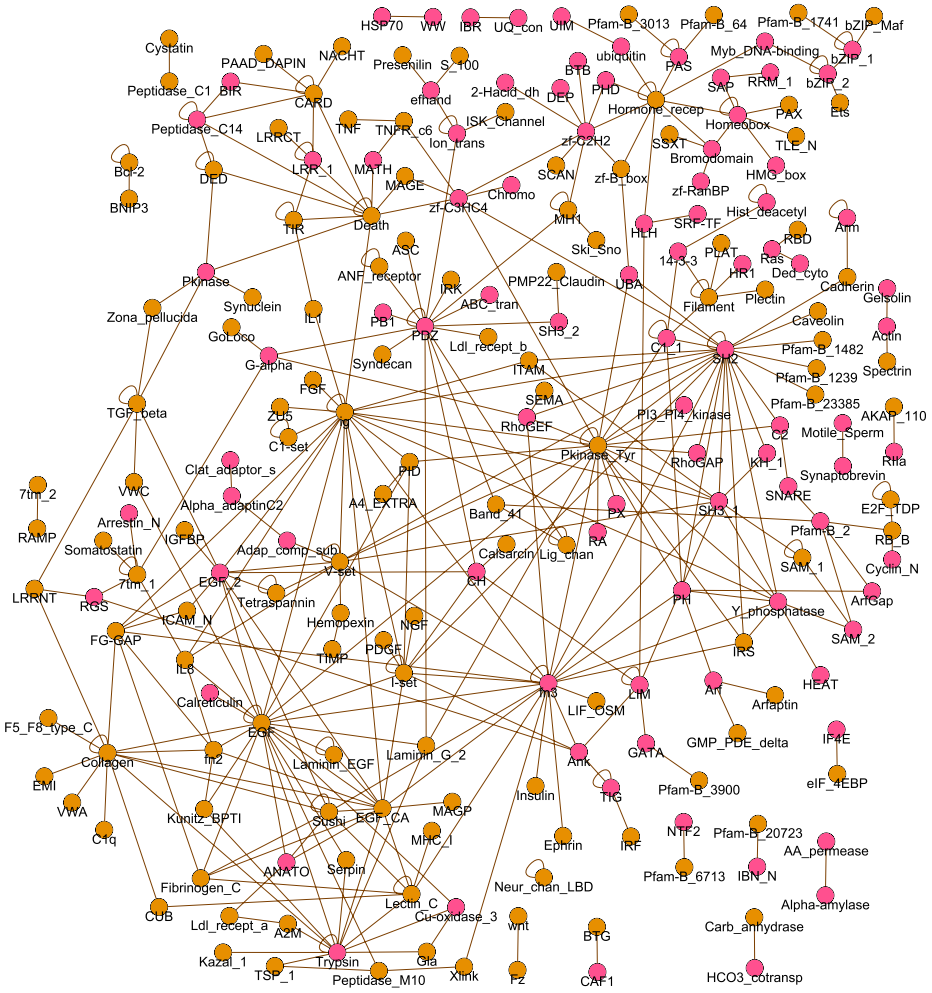

Figure 2.10: Pfam pair schemas that are emergent in *H. sapiens* and do not have instances in the *S. cerevisiae* interactome. Red vertices indicate Pfam motifs that are found in both organisms, and brown vertices indicate Pfam motifs found in *H. sapiens* but not *S. cerevisiae*.

respectively. *H. sapiens* emergent schemas that have instances in *S. cerevisiae* complete this portion of the pathway by connecting the ubiquitin family with the *UQ_con* family of ubiquitin-conjugating proteins. Additionally, *H. sapiens* schemas connect *ubiquitin* to the *HSP70* family of chaperones, reflecting the role of ubiquitination in targeting misfolded proteins for degradation.

2.3.7 Schemas enable functional predictions

There are several motifs of unknown function implicated in schemas (e.g., Pfam-B motifs in Figures 2.5-7 and Figure 2.9). As proof of concept, we focus on two examples, *DUP* and *MAGE*, and show that schemas can help characterize motifs and proteins whose functions have not yet been experimentally determined.

"One of the most curious gene families in yeast" [65], the *DUP* family consists of twenty-three yeast proteins [12], most of which are not yet functionally annotated. Based on schema analysis, we predict that the *DUP* family consists of proteins that are associated with membrane transporters. The *DUP* proteins are found in multiple schemas of various topologies (Figures 2.5, 2.6, and 2.7), and these schemas are dominated by interactions with members of transporter families such as *MFS_1*, *Sugar_tr*, and *AA_permease*. The finding that one member of the family, Cos3, is an enhancer of the antiporter Nha1p [51] supports this prediction. Additionally, a previous prediction connects *DUP* proteins with membrane trafficking [12]; given our analysis, they might be involved in trafficking of transporters.

There are fifty-five *MAGE* sequences in *H. sapiens* [9], thirty-two of which are listed as such in Pfam and nine of which have physical interactions listed in BioGRID [7]. *MAGE* proteins, which are mostly uncharacterized, were initially found to be expressed in tumors, although some are now known to be expressed in normal tissues. We found the *MAGE* family to participate in pair schemas with two protein families: the *Death* domain and the *zf-C3HC4* RING motif (see Figures 2.10 and 2.11). The *Death* domain is associated with apoptosis, and the RING motif is associated with E3 ubiquitin ligases, which perform the final step in protein ubiquitination. These schemas suggest a connection between *MAGE* proteins and apoptosis, which, if correct, could shed light on the association between some of the original members of the *MAGE* family and cancer. It is possible that ubiquitination plays a role in this connection, although the link between ubiquitination and apoptosis is still

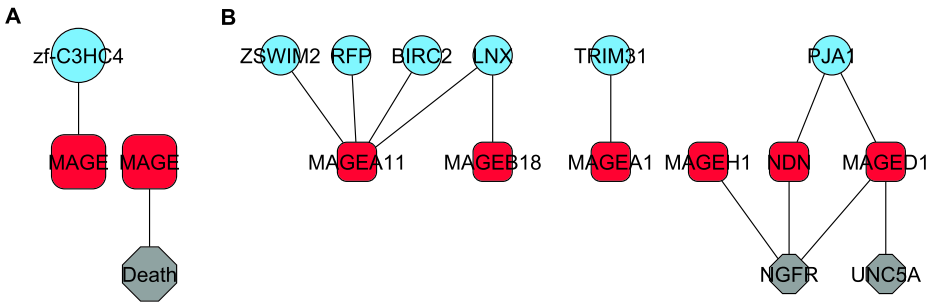

Figure 2.11: Emergent *H. sapiens* pairs involving the *MAGE* family (**A**), and their instances (**B**).

a subject of investigation; *MAGE* proteins may provide a connection between these two processes. We further note that the *zf-C3HC4* RING domain forms a schema with the *Death* domain as well (Figure 2.10).

2.4 Discussion

We have introduced network schemas as a general means to describe organizational units consisting of particular types of proteins that work together in biological networks and have developed a fully-automated procedure for discovering them. In the first analysis of this type, we have uncovered hundreds of emergent network schemas and have demonstrated that they recapitulate known biology, suggest new organizational units, have enriched biological process coherence, and have instances in organisms across large evolutionary distances.

Using two poorly understood gene families, one from human and one from yeast, we have shown how schema analysis can be used to annotate protein families and their individual members. Guilt-by-association and other network-based functional annotation methods (review, [72]) are, by intent and design, better suited for the general function prediction problem. However, schema analysis provides a new way to amplify a weak signal, and can suggest mechanistic details in some cases. For example, if we consider proteins that interact physically with a given protein, and

then take the most over-represented biological process annotation among them using the GO Generic Term Finder [6], we find "apoptosis" as a prediction for only two of the *MAGE* proteins having physical interactions (at corrected p-value ≤ 0.05 level) and no ubiquitination related predictions. On the other hand, schema analysis of the *MAGE* proteins acts as a lens that focuses the investigator's attention on patterns of interaction that together are statistically significant.

The prominence of emergent schemas related to signaling suggests that we may be able to utilize them to uncover pathways. Previous approaches to predicting signaling pathways from protein physical interaction networks have attempted to find paths from receptors to transcription factors [68, 81], and then evaluating them (e.g., based on gene expression coherence [74,81]). Alternate approaches have attempted to query interactomes in order to find pathways homologous with known pathways [37]. Schemas may instead be used in pathway discovery by restricting or favoring paths in a network based on schema annotations, or using schemas to evaluate or score the enumerated paths. Indeed, simply by taking overlapping emergent network schemas and obtaining their instances in the full unfiltered *S. cerevisiae* interactome, we can recover portions of known pathways. For example, by considering just the triplet schemas *RhoGAP-Ras-RhoGEF*, *RhoGAP-Pkinase-Ras*, and *Ras-Pkinase-SH3_1*, we obtain significant portions of the cell wall organization and biogenesis and cell polarity pathways, and the related pathways of filamentous growth and pheromone response, as well as the cell cycle and vesicle transport pathways.

Our results can be considered in terms of several alternate hypotheses concerning the evolutionary processes by which schemas arise. Did the different instances of a schema arise from a common ancestral group of interacting proteins which then proliferated, or did convergence play a role? It is likely that both processes took place, with one or the other being more important in different schemas. In the case of Pfam schemas, this question is on the one hand analogous to, and on the other hand

intimately related to, the question of how intra-protein domain architectures arose (e.g., see [5,16]). As a result, the possible role of domain duplications, insertions and shuffling is an important consideration in understanding the evolutionary histories of individual Pfam schemas. For example, in the case of intra-protein domain architectures, graph-theoretic analysis has suggested that combinations involving certain promiscuous domains (SH3 and C2, among others) may have arisen more than once, though other combinations may be the result of the formation of a single ancestral sequence that proliferated through duplication [61]. For schemas that are based on protein annotations that do not necessarily arise from sequence similarity (e.g., GO molecular function schemas), convergence is likely to play a larger role, as the proteins comprising distinct instances may not share any discernable sequence similarity.

Another question that arises is how novel schemas are incorporated over the course of evolution. A comparison of emergent pair schemas in *S. cerevisiae* and *H. sapiens* provides some hints, but further analyses of the interactomes of many organisms is necessary to obtain a better understanding. Similarly, what is the relationship between emergent and non-emergent lower-order schemas that together make up a higher-order emergent schema? Was the non-emergent component added to the earlier emergent one? The techniques introduced in this paper provide a computational foundation for the extensive cross-genomic studies that are necessary to attempt to address these and related questions.

Depending on the intended application it may be desirable to modify the computational procedure for uncovering emergent schemas. The described approach is designed to be conservative in several respects. First, since we search for proteins that work together in a specific topological pattern, we use only networks comprised of direct physical interactions, erring on the side of caution in the case of pull-down data. Alternate approaches may instead be taken to enrich the number of direct interactions but not exclude other types of interactions [39]. Second, we require each

emergent schema to have at least two independent instances. Interesting schemas certainly get excluded as a result (e.g., several SCF ubiquitin-ligases in *S. cerevisiae* that differ only in their F-box protein component [58]). Nevertheless, independence helps ensure that an emergent schema is truly recurring and that it does not depend on the occurence of any single interaction; this is an important consideration due to the underlying noise in the network [79]. Finally, we search for emergent schemas bottom-up, eliminating schemas that may owe their significance solely to the significance of their lower-order constituents; this favors including lower-order schemas over higher-order ones. It is possible, however, that in some cases, the higher-order schema is the recurring working unit that makes its lower-order components look significant. Our schema-finding procedure can be modified to relax any of these requirements, and indeed we believe that there are many more functionally important and recurring schemas than we have identified here.

In this work we have examined four of the most basic topologies for schemas. However, additional or flexible topologies (e.g., allowing optional proteins) may also be considered. The primary challenges in extending our current approach lie in computationally enumerating all possible schemas and in developing effective algorithms for maintaining the distribution of the appropriate lower-order constituents. Additionally, whereas here we have considered annotations consisting of Pfam motifs and a subset of GO molecular function terms (each separately), schemas based on several complementary systems of protein labels that annotate at differing levels of resolution may provide a more multidimensional view of protein function; in this case, the hierarchical relationships between annotations would need to be better handled.

A noticeable feature of our analysis is that the underlying data treats all interactions as being the same. In reality, the interactions have both meaning and contextual information. For example, some schemas consist of interactions representing the (de)activation of one of the interactors by the other, with corresponding temporal in-

formation. A triplet schema, for example, may correspond to a central protein acting upon its two spoke proteins, or two spoke proteins acting upon the central protein, or one spoke protein acting on the central protein which then acts on the other spoke protein. Schemas may also include a combination of multiple subschemas that are active at different times or in different cellular contexts. Such information is not explicitly present in the schemas we have uncovered and is an especially important consideration when studying multicellular organisms, in which different interactions may take place in different cell types altogether. If contextual information for a large number of interactions becomes known and systematized, it is possible to look for schemas either within each context separately, or include contextual information as part of the schema definition. Alternatively, one could attempt to extract contextual information from the current schemas, focusing on the individual undirected schemas that our approach presently finds, and devising computational means for predicting such information based, for example, on expression information or literature search. Such inclusion of information about the biological context of when interactions occur should refine the network schemas observed. Moving beyond physical interactions, an interesting avenue for future work would be to extend network schemas to specify other types of interaction as well, as has been done for network motifs [62, 93]; the "meaning" or semantics of these types of network schemas would be very different from the type considered here. Schemas uncovered in one type of network can also be used to interrogate other networks. For example, schemas from a physical interaction network may help identify direct interactions in functional networks for organisms for which no large-scale physical interactomes have been determined.

Finally, while here we have searched for emergent schemas in just two sample organisms, our techniques can be applied to a greater number of interactomes across the evolutionary spectrum. This would enable us to uncover what types of schemas

are found in different organisms, and to better address how networks expand or change to incorporate new motifs or protein functions.

Chapter 3

NetGrep: fast network schema searches in interactomes

3.1 Introduction

In the previous chapter, we formalized network schemas as a means for representing organizational patterns within interactomes, developed a fully automated system for uncovering over-represented schemas in physical interactomes, and applied it on the *S. cerevisiae* interactome. A key component of that computational pipeline, given a particular network schema, is to extract its matches from an interactome. In this chapter, we describe the fast network schema search algorithms that enabled the analysis previously presented. In particular, we present a general system, NetGrep, that integrates the wealth of prior information about individual proteins—e.g., their functional annotations, sequence motifs, predicted domain structures, or other attributes—within the context of fast, user-directed network schema searches within biological networks consisting of heterogeneous interaction types.

The NetGrep system allows querying with schemas described via a diverse set of protein features, including Prosite family [32], Pfam motif [4], SMART domain [47,67],

Supfam superfamily [21], and Gene Ontology (GO) [2] annotations. Proteins may also be specified via particular protein IDs, homology to other proteins, regular expressions over amino acids, or with unions or intersections over any of the previously described features. By utilizing these protein attributes in combination with physical, genetic, phosphorylation, regulatory, and/or coexpression interactions (as available for the organism of interest via high-throughput experiments), the network schema queries allowed in NetGrep generalize many previously studied interaction patterns, including domain-domain interactions, signaling and regulatory pathways, and more complex network patterns. For example, a general network schema relating to signaling is a path of physically interacting proteins, where the first protein is a receptor, and the last protein is a transcription factor (Figure 3.1A); such queries have been used in conjunction with gene expression data to infer signaling pathways in *S. cerevisiae* [81]. A more specific network schema relating to signaling consists of particular proteins making up a pathway which can be used to search for paralogous pathways (Figure 3.1B), as has been suggested in network alignment approaches [38]. Network motifs have been widely studied [50, 73], and can be described by schemas without constraints on protein types but with particular interaction types specified (Figures 3.1C and 3.1D). Domain-domain or domain-peptide interactions, such as those important for cell signaling and regulatory systems [59], can be represented by two-protein schemas with the proteins appropriately constrained (Figure 3.1E). Schemas relating to specific proteins of interest are also easily incorporated (Figure 3.1F). Finally, network schemas can be naturally extended to handle approximate matches by specifying optional nodes (Figure 3.1A). While these types of network interaction patterns have been studied in a wide-range of contexts, it has not even been possible to use many of them as queries in existing systems. Thus, we have introduced NetGrep to provide a flexible, unified system for interrogating an interactome using a diverse set of queries.

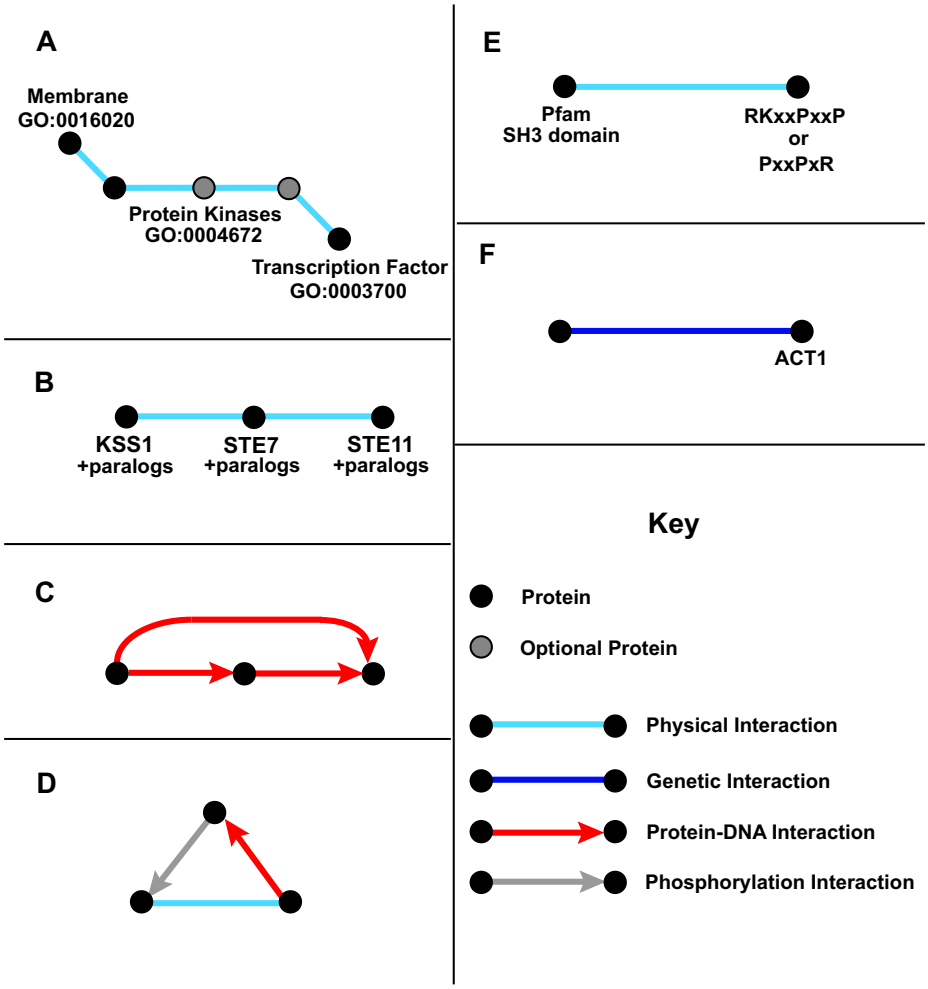

Figure 3.1: Examples of network schemas. Unlabeled schema proteins are considered to be 'wildcards' and can match any protein in the interaction network. **(A)** A signaling pathway schema. This schema matches all sets of proteins such that a protein in the cell membrane physically interacts with a succession of anywhere between one and three kinases, the last of which physically interacts with a protein that is a transcription factor. **(B)** A MAP kinase schema, specified by particular yeast proteins making up a canonical MAPK signaling pathway. **(C)** A feed-forward loop network motif [50] schema. The unlabeled nodes can match any protein in the network. **(D)** A 'kinate' feedback loop network motif schema [62] **(E)** An SH3 domain interaction schema. This schema matches all interacting pairs of proteins such that one contains a Pfam SH3 domain and the other has one of the specified patterns, corresponding to SH3 binding sites, in its underlying amino acid sequence. Amino acids in the pattern are specified by their one letter code, and 'x' denotes a match to any amino acid. **(F)** A specific protein schema. This schema matches all proteins with a synthetic lethal relationship to yeast protein ACT1.

In addition to allowing a broad range of network schema queries, NetGrep has an easy-to-use graphical interface for inputting schemas. For each user-input schema, NetGrep finds all of its matches in the chosen interactome. Although the search problem is a case of the computationally difficult subgraph isomorphism problem, we have been able to develop algorithms that take advantage of schema characteristics for biological networks. As a result, NetGrep's core algorithms are extremely fast in practice for queries with up to several thousand matches in the interactomes studied. Though speed is useful for individual user queries, it also makes it possible to systematically enumerate and query many network interaction patterns. For example, here we have systematically tested NetGrep's underlying algorithms by enumerating $> 100,000$ schema queries with proteins described via GO molecular function terms and have found that for schemas with up to tens of thousands of matches, NetGrep can rapidly uncover all instances. Our algorithms can thus enable new analysis that characterizes networks with respect to the types and numbers of interaction patterns found (e.g., see Chapter 2).

3.1.1 Relationship to previous work

There are several previously developed tools for querying biological networks, although none of them have the full functionality of NetGrep. Previous approaches fall broadly into the categories of network alignment, network motif finding, and specific subgraph queries, although these categories overlap.

Network alignment tools [15, 36, 38, 42] align protein-protein interaction networks by combining interaction topology and protein sequence similarity to identify conserved pathways. These tools can be used to identify schemas for which the criterion for matching a query protein to a target protein is sequence similarity. Network alignment has also been applied to metabolic networks [60], with proteins chacterized by their enzyme classification. Algorithmically, these approaches are designed for align-

ing entire interactomes, and several of them are based on local alignments based on simpler linear or tree topologies. NetGrep in contrast is developed and optimized for general network schema queries, and has faster algorithms for the task at hand.

Several tools exist for uncovering network motifs or over-represented topological patterns in graphs [66, 87], and these could be used to find schemas consisting solely of unannotated proteins. These approaches do not, however, provide a mechanism for utilizing specific protein annotations, nor do they allow user defined queries. We note that while NetGrep can obtain instances to network motif queries, our algorithms are optimized for schemas utilizing protein descriptions and with up to tens of thousands of instances. Alternate algorithms, specifically developed for counting or approximating the total number of instances of network motifs [1, 22], may be more suitable if network motif queries are desired.

Other more closely related tools have been implemented to query biological networks using subgraphs. Given a linear sequence of GO functional attributes, Narada [56] finds all occurrences of the corresponding linear paths in a network. MOTUS [45] is designed for non-topology constrained subgraph searches in metabolic networks. Qnet [13] is restricted to tree queries and utilizes only sequence similarity. NetMatch [14], extending ideas of GraphGrep [19], allows users to search for subgraphs within the Cytoscape [69] environment and can be used for simple schema queries. SAGA [83] is a subgraph matching tool for Linux platforms that allows inexact matches to a query in multiple networks, and has built-in support for biological networks where proteins are described via orthologous groups. In contrast to these approaches, NetGrep is a standalone, multi-platform system where schemas may have arbitrary topologies as well as a large set of built-in protein and interaction types. NetGrep schemas allow flexibility via optional nodes (thereby permitting inexact matches) and protein and interaction descriptions that may consist of boolean conjunctions or disjunctions of features. While NetGrep comes with built-in protein

Feature	PathBlast [38]	Fanmod [87]	Narada [56]	Saga [83]	NetMatch [14]	NetGrep
Non-linear queries	X	X		X	X	X
Allows arbitrary protein annotations		1/node			X	X
Boolean combination of annotations				X		X
Inexact matches	X			X		X
Multiple edge types in a network		X			X	X
Boolean combination of edge types						X
UI to search/choose annotations			X			X
Can be used with Cytoscape					X	X
Can be used as a standalone	X	X	X	X		X
Custom data sets provided	X			X		X

Table 3.1: A comparison of built-in features available in systems that can in principle be used for querying interactomes using network schemas. A network alignment tool, PathBLAST, and a network motif finder, Fanmod, are shown for comparison. All other systems are explicitly designed for querying interactomes utilizing labeled subgraphs.

feature and interaction data sets for several model organisms, it also has the ability to incorporate new custom networks and associated feature sets. Furthermore, NetGrep can optionally be used within the Cytoscape environment to visualize schema matches. See Table 3.1 for a comparison of features available in NetGrep and previous approaches.

3.2 Implementation

We have implemented NetGrep in Java so that it is easily portable among different operating systems. Users have the option of running a feature-limited version of the software on our server (located at http://genomics.princeton.edu/singhlab/netgrep/) or of downloading the fully featured program and running it locally. NetGrep can be used both as a standalone application or in conjunction with Cytoscape as a plugin if visualization of the results in network form is desired. A detailed description of how

to use NetGrep is provided online. More formal descriptions of schemas, their instances in the interactome, and the algorithms used to uncover the instances are given in **Model and Algorithm** below.

3.2.1 Packaged data files

Data files are provided for the following model organisms to be used with NetGrep: *S. cerevisiae, C. elegans, D. melanogaster* and *H. sapiens*. These files contain all the information necessary to run NetGrep, including protein information (names and aliases), interaction maps, and protein features.

Physical and genetic interactions for all organisms are obtained from BioGrid [7] (version 2.0.34), and phosphorylation interactions for yeast are obtained from [62]. Regulatory relationships in yeast are obtained from the binding data of [25] using a p-value/cutoff of 1e-5. Gene expression interactions between pairs of proteins are taken as those that have linear correlation coefficient > 0.8 on the concatenation of all experiments in the gene coexpression data compiled by [82]; we note that this high cutoff and required correlation in all conditions favors expression interactions between housekeeping proteins.

One important feature of NetGrep is that none of the data is hard-coded into the program. Users can therefore use any node features or edge types desired when constructing networks; for example, custom or newly defined interaction types can be added. Additionally, creating data files for other, non-supported organisms is a straightforward process.

3.2.2 Describing proteins and interactions

Nodes, describing proteins, are added to a schema via a visual canvas, and then individual features of the proteins can be selected (Figure 3.2A). The interactome to be queried is specified via a pull-down menu (Figure 3.2B). Each of the nodes in a

schema can be annotated with any combination of protein features; multiple features are related by boolean combinations via ANDs or ORs. A node in a schema can be connected to any other, corresponding to a desired interaction, also by specifying this in the visual canvas. These edges between nodes can be described as having one or more types (Figure 3.2C). As with protein features, edge types may be combined with logical ANDs or ORs. For example, one might require that two given proteins physically interact AND that the first is a transcription factor regulating the second. Note that a schema must be a connected graph.

3.2.3 Specifying inexact matches

The schemas described thus far are rigid in their structure. Occasionally, a user might prefer to specify that any number of proteins with a particular feature set interact in a cascade or that a given node in the schema not be absolutely required. NetGrep achieves this flexibility by allowing nodes in the schema to be designated as optional. When a schema contains an optional node, NetGrep will find matches both with and without the given protein. For example, to represent a signaling pathway as "a protein in the membrane, which interacts with a succession of between one and three kinases, the last of which interacts with a transcription factor," one would build the given linear five-node pathway and designate two of the kinases as optional (Figure 3.1A). NetGrep would then find all three, four, and five-node matches within the network. Note that single nodes with more than two interactions cannot be designated as optional. When an optional node has two interactions, the interaction types are logically ORed for instances of the schema that have the optional node excluded.

Similarly, a significant problem with current interaction datasets is that they are incomplete. NetGrep provides a solution to this difficulty by also allowing interactions in a schema to be designated as optional. When a schema contains an optional

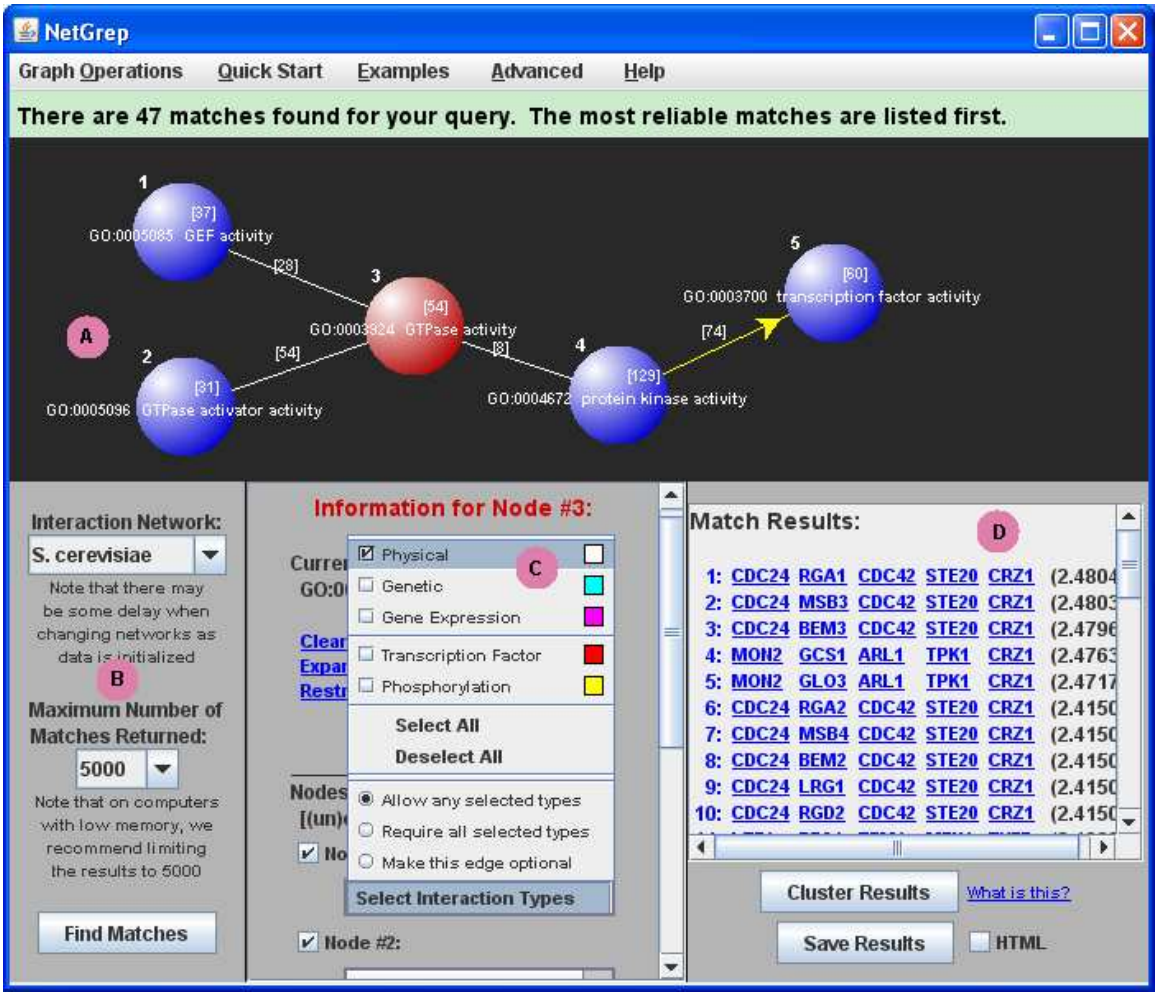

Figure 3.2: A detailed screenshot of the NetGrep display showing a sample query schema. **(A)** The graph panel area used to describe schemas. The Ras GTPase signaling schema from Figure 2.1 is shown in the panel with the Ras GTPase node highlighted. **(B)** The panel used to designate which interaction network to use, to choose the maximum number of matches desired, and to initiate a search. **(C)** The panel used to annotate nodes in the schema and to create or modify edges. The information for the highlighted node (node #3) is currently displayed in the panel; the edge between the first and third nodes is being modified. **(D)** The results panel in which the matches found from the search are displayed. Each row lists the proteins which make up a particular match along with its reliability score.

interaction, NetGrep will allow matches even if the given interaction is not found in the network.

3.2.4 Matches and reliabilities

NetGrep has a user-set threshold that limits the number of matches reported for an input schema (Figure 3.2B). As a typical user is not expected to look through tens of thousands of matches, this threshold can be as low as 100 and as high as 50,000. For faster run times, a lower threshold is recommended; additionally, the threshold limits memory usage. Alternatively, if the total number of instances is greater than the highest allowed threshold, there is an advanced (somewhat slower) option that computes the total number of instances but does not explicitly enumerate them.

The instances of a query schema are returned by NetGrep, up to the user-defined threshold, and are sorted according to how confident we are of the underlying interactions. In particular, for each pair of proteins, we have a single precomputed reliability value between 0 and 1 that assesses how likely these two proteins are to interact (see **Interaction Reliabilities** below). For each of the matches found by NetGrep, its overall reliability is computed by multiplying together the reliabilities corresponding to protein pairs that have interactions in the matches. The matches are sorted based on the negative log of this value, beginning with the most reliable (Figure 3.2D).

3.3 Model and Algorithm

3.3.1 Graph Model

We give a formal specification of the problem. Let $\mathcal{L}$ be the set of possible protein labels (e.g., Pfam motifs, protein IDs, etc.) and let $\mathcal{T}$ be the set of possible edge types (e.g., physical, regulatory, etc). An interaction network is represented as a mixed graph $G = (V_N, E_N, A_N)$. V_N is the set of *vertices*, with a vertex $v \in V_N$ for each protein. $E_N \subseteq V_N \times V_N$ is the set of undirected edges, and $A_N \subseteq V_N \times V_N$ is the set of arcs or directed edges. Vertices correspond to proteins and edges and arcs

correspond to interactions. Each vertex v in the interaction network is associated with a set of features $l(v) \subset \mathcal{L}$ (specifying protein features), each edge (u, v) is associated with a set of types $t_e(u, v) \in \mathcal{T}$ (specifying the undirected interactions between the proteins), and each arc (u, v) is associated with a set of types $t_a(u, v) \in \mathcal{T}$ (specifying the directed interactions between the proteins). If there is no edge between u and v, $t_e(u, v) = \emptyset$, and if there is no arc between u and v, $t_a(u, v) = \emptyset$.

A *network schema* is a mixed graph $H = (V_S, E_S, A_S)$ such that: (1) each vertex $v \in V_S$ is associated with description set $\mathcal{D}_v$ such that each $d \in \mathcal{D}_v$ is a subset of $\mathcal{L}$. In NetGrep, the set $\mathcal{D}_v$ is constructed via individual protein features in $\mathcal{L}$ and utilizing either intersections or unions over these features. For example, for a particular vertex $v \in V_S$, if a union is taken over individual feature types, $\mathcal{D}_v$ consists of singleton sets consisting of each of these features. Note that $\mathcal{D}_v$ can consist of one set, the emptyset, in the case of a wildcard vertex. (2) for every pair of vertices u and v such that $(u, v) \in E_S \cup A_S$, there is an associated description set $\mathcal{D'}_{u,v} \subset \mathcal{T}$. In NetGrep, the set $\mathcal{D'}_{u,v}$ is constructed via individual interaction types, and requiring either all of them, or just one of them. For example, for a particular pair of vertices u and v with desired edges or arcs between them, if all interactions are required, then $\mathcal{D'}_{u,v}$ consists of a single set consisting of all desired interaction types.

An *instance* of a network schema H in an interaction network G (i.e., a match in the network for the schema) is a subgraph (V_I, E_I, A_I) where $V_I \subset V_N$, $E_I \subset E_N$, and $A_I \subset A_N$ such that there is a one-to-one mapping $f : V_S \rightarrow V_I$ where (1) for each $v \in V_S$, there exists a $d \in \mathcal{D}_v$ such that $d \subset l(f(v))$ (2) for each pair of vertices $u, v \in V_S$ with $(u, v) \in E_S \cup A_S$, there exists a $d' \in \mathcal{D'}_{u,v}$ such that $d' \subset (t_e(f(u), f(v)) \cup t_a(f(u), f(v)))$. Note that two distinct instances of a schema may share proteins and/or interactions; however, any two instances must differ in at least one protein. Network schemas are used to interrogate the interaction network for sets of proteins which match this description.

3.3.2 Interaction Reliability

For each pair of proteins, we estimate the reliability of their having any interaction between them. In particular, we first partition all the observed underlying interactions in the interactome into several experimental groups. The reliability of each experimental group i is then evaluated as follows. For experiments determining non-genetic interactions, the reliability is estimated based on "functional coherence" by computing s_i as the fraction of interactions in that group that are between proteins sharing a high-level GO biological process slim term [26] (only pairs of interacting proteins that both have GO slim annotations are considered). We note that we do not use the functional coherence measure to assess genetic interaction experiments, as these types of interactions can bridge between pathways [85]. Instead, for these experiments, the reliability is estimated based on a "2-hop" topological measure that has been shown to be highly predictive of genetic interactions [89]. In particular, the reliability s_i for an experimental group determining genetic interactions is estimated by computing the fraction of interactions in that group that additionally have paths of length two between them in the full interactome where either both interactions are genetic interactions or where one is a genetic interaction and the other is a physical interaction. Then, for a pair of proteins u and v, we consider all interactions j found between them, and treat them as independent events. The reliability $r(u, v)$ between u and v is then computed as

$$r(u, v) = 1 - \Pi_j(1 - s_{g(j)}),$$

where j ranges over all interactions linking proteins u and v, and $g(j)$ gives the experimental group of interaction j. If no interactions exist between the two proteins, $r(u, v) = 0$. This noisy-or scheme is similar to the one used for reliability estimation in [52, 86].

We partition our interactions into the following experimental groups. For physical and genetic interactions, there is one group for each individual high-throughput physical and genetic interaction experiment (defined as those that discover at least 50 interactions). All small-scale physical interaction experiments (defined as those that discover fewer than 50 interactions) are considered as belonging to a single group. Similarly, small-scale genetic interaction experiments are considered a single group. Experiments are identified by the combination of "Experimental System" and "Pubmed ID" as reported by the BioGRID [7]. All phosphorylation interactions in [62] are considered in one group. In the case of interactions that are associated with continuous numerical data, such as coexpression interactions (associated with the correlation coefficient) and regulatory interactions [25] (associated with the p-value for the binding), we assign each interaction to one of 20 uniform bins associated with the numerical data, and consider each bin as a separate group.

3.3.3 Searching for schemas

Overview

Finding the matches for a particular schema in a network corresponds to the computationally difficult subgraph isomorphism problem. A number of sophisticated algorithmic approaches for closely related problems on biological networks have been introduced earlier (e.g., utilizing color coding [13]). Here, we obtain fast matches in practice utilizing a few key ideas. First, we pre-process the interactome to build fast look up tables mapping protein and interaction type labels to proteins associated with the labels. For each node in a schema, this allows us to quickly enumerate the set of all proteins which match the nodes's feature set. Second, we utilize the labeled schema nodes and schema edges to prune the search space. In particular, we constrain the proteins in each node match set by determining interaction matches along each edge in the schema. Finally, these interactions are cached for fast lookup in the last

step, in which we enumerate the considerably smaller search space, and construct the full list of matches. We describe these steps in more detail below.

Algorithm

We first pre-process the interactome to maintain two hashes which map labels to proteins associated with those labels. $HASH_F$ maps protein features to sets of vertices described by those features (e.g., all kinases), and $HASH_T$ maps edge types to pairs of proteins connected by an edge annotated with the types (e.g., all proteins with physical interactions). For directed edge types, there are two separate entries in $HASH_T$, one for each direction of the edge (e.g., one for all kinases and one for all substrates). These hashes are used to quickly build, for any schema, its matches edge by edge.

When searching for instances of a particular schema, we associate with each node v in the schema a set of node matches $NMATCH_v$, which contains all of the proteins in the interaction network which are described by that particular schema node (i.e., the proteins that could be a match to that schema node). Specifically, we use $HASH_F$ to initialize $NMATCH_v$ with all the proteins that match v's feature set. When features are combined with a boolean AND, we take the intersection of the protein sets from $HASH_F$, and when they are combined with a boolean OR, we take the union of the protein sets. For each edge $e = (u, v)$ in the schema that has a single type (i.e., is not comprised of a boolean combination of types) or for which all edge types are required (i.e., types are combined by a logical AND), we use $HASH_T$ to trim the proteins in each node match set. For example, if schema node v is connected by a physical edge, then we can remove all proteins from $NMATCH_v$ which are not found in the set from $HASH_T$ corresponding to all proteins in the network connected by a physical edge.

We next prune the sets of node matches as follows, or until any of them becomes empty (at which point we know that there are no matches to the query in the network). For each edge $e = (u, v)$ in the schema, we use the network interaction map to remove all proteins from $NMATCH_u$ which do not interact with any of the proteins in $NMATCH_v$ given e's specified type. Although we could repeat this pruning step after each edge is processed, we have found it to be unnecessary because of two additional optimizations that we introduce. First, as we iterate through the edges in this step, we start with those edges whose endpoints contain the smallest sets of node matches and we progress in order; this optimization helps to reduce the size of the larger node match sets early on in the process. That is, we rank schema nodes based on the size of their node match sets, start with the node with the smallest node match set, and consider its edges first, starting with the neighbor with the smallest node match set. We then consider the node with the next smallest node match set, and so on. Second, as we iterate through the schema edges, we cache the matches for each edge, so that they can be quickly accessed in the next step where we find the actual matches. Note that this pruning step is skipped with optional nodes because edges connected to those nodes are not required. This pruning step is also skipped for edges if their match bins are too large (> 1000).

To find the sets of proteins that match the given schema, we iterate through each of the node match sets from smallest to largest, constructing matches as we go along. We note that this search order over the nodes provides a significant speed-up over a simpler approach that performs depth-first search from an arbitrary starting node in the schema. As we iterate through the nodes, for each protein p in a given match set representing node v in the schema, we constrain each *larger* match set representing node u in the schema as follows: if u and v are connected by an edge in the schema, we eliminate all proteins in u's match set that do not interact with p (using the cached matches from the pruning step above). Furthermore, we remove p from u's set if it

is there (i.e., we do not allow the same protein to occur in multiple positions of a match). We then set p as the matching protein at schema node v for this particular set of matches and traverse to the next largest node match set. Once a complete match to a schema is found, we backtrack and continue the search process.

If at any point the number of matches to a schema exceeds the user-defined threshold (Figure 3.2B), the search is terminated and NetGrep returns just those matches found up to that point. Once all matches to a schema are found, they are sorted by their interaction reliability, as described above.

Symmetric schemas

When a schema displays an inherent symmetry, it is often the case that the same set of proteins redundantly occurs in multiple instances. Consider, for example, the symmetric linear three-node schema A-B-A, where the edges are undirected, and the first and last nodes have identical feature sets and are symmetric around the middle node. One might find among the matches of this schema the proteins p_1-p_2-p_3 and p_3-p_2-p_1. NetGrep is able to determine that a given schema is symmetric and excludes these superfluous matches from the results returned by the search. The test for symmetry exploits the fact that for any two given nodes in a schema to be symmetric they need to have the exact same feature set and degree; for all pairs of nodes u and v in the schema for which this is true, the algorithm recursively checks all pairs of nodes connected to these two target nodes (i.e., one connected to u and one connected to v) for symmetry, following any given edge just one time. This is equivalent to a depth first search over the schema. The base case in the recursive algorithm occurs when two target nodes are connected to each other or when they are connected to the same node.

If a query is determined to be symmetric, redundant matches are ignored during the search. To accomplish this task, each protein in the interaction network is first

assigned an arbitrary unique ID number, as are each of the nodes in the query schema. Then, for any two symmetric nodes A and B in a query schema where the ID of A is smaller than the ID of B, we require that the ID of any protein matching node A be smaller than the ID of a protein matching node B in any given instance. All instances for which this requirement is not met for each of the symmetric nodes are ignored.

3.3.4 System Requirements

The NetGrep system is implemented in Java and has been tested on Windows, MacOS, and Linux. It requires Java 1.5 or higher to run, and the source code is available open source with a GNU public license.

3.4 Performance

We have found NetGrep to run extremely fast in practice. We illustrate the performance of NetGrep in two ways. First, we report how long NetGrep takes for each of the schemas shown in Figure 3.1. As a comparison, whenever possible, we have also run these schemas on the same network using other tools. For each system, the software is downloaded and run on a laptop running Windows XP with 1GB RAM and a 1.66GHz Intel processor. All queries are run on our *S. cerevisiae* network data, described above. All timings include the times for both the search and output of the results. Default settings for all programs are used. While we have NetGrep print out its wall clock time to standard output, the timings for the other systems are estimated via a handheld timer and rounded down to the nearest second. We have chosen this process as some of the systems must be run within a graphical interface and strict system timing calls are not possible. Each query is repeated ten times and the reported running times are the averages over these runs. Table 3.2 shows the performances for each sample query. Note that table entries are left blank for

Sample Query	PathBLAST	Fanmod	Narada	NetMatch	NetGrep
Signaling pathway #1			28		4.2
Signaling pathway #2					26.9
MAPK pathway	90				0.02
Feed-forward motif		32		5.2	1.4
Kinate motif		32		5	0.5
SH3 domain interaction					0.5
ACT1 genetic interaction				15	0.1

Table 3.2: Running times (in seconds) for several sample queries on the *S. cerevisiae* interaction network, using PathBLAST, Fanmod, Narada, NetMatch and NetGrep. All reported running times are for search and output only. As in Table 3.1, PathBLAST is used as a prototypical example of a network alignment tool and Fanmod represents network motif finders. Note that SAGA is excluded here because it cannot be run on Windows. The sample schemas correspond to those provided in Figure 3.1, except that two distinct queries are used for Figure 3.1A. In the first, all three kinases in the pathway are required. In the second, two of the kinases are designated as optional (as in Figure 3.1A). Each query is run ten times and the average computation time is provided. Row entries are left blank for any tool which is unable to find instances of a particular schema because of feature limitations.

schemas which cannot be run on a given system, and two of these queries can currently be run only on NetGrep. NetGrep has considerably faster query times for all sample queries, and is often more than an order of magnitude faster than previous approaches.

Second, we have run NetGrep in a systematic fashion on schemas consisting of physical interactions in triangular, 4-node linear "quad," and 4-node branched (i.e., a central node interacting with three others) "Y-star" topologies. We consider all possible ways to annotate the proteins in these topologies using GO molecular function slim [26] terms. We have chosen these types of schemas because of their linear, branched, and cyclical topologies, and because we are easily able to exhaustively enumerate over all possible schemas of this type on a standard laptop. Additionally, GO annotations can be utilized with queries in two previous systems, NetMatch and Narada (though Narada is limited to the linear schemas). There are 1,771 triangular schemas, 101,871 quad schemas, and 37,191 Y-star GO molecular function slim schemas. Since each GO slim term is general and can annotate many proteins, we set the threshold for the maximum number of matches allowed to 80,000. Of the

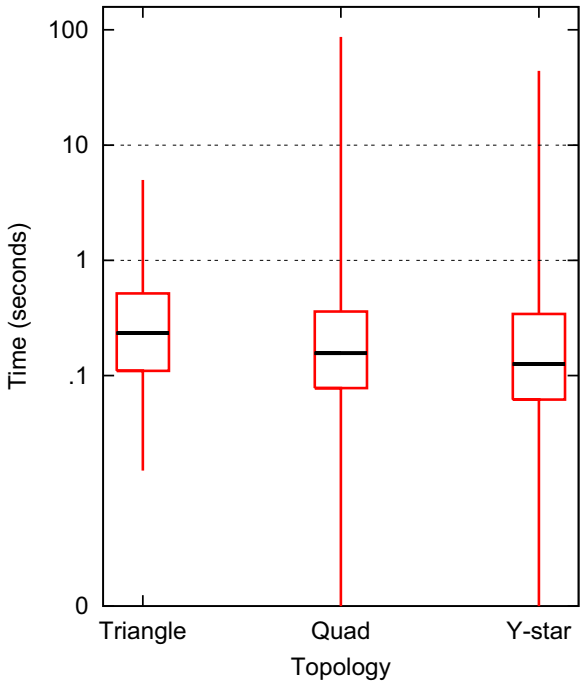

Figure 3.3: All possible triangular, 4-node linear, and 4-node branched schemas ('Y-star') with nodes described via GO molecular function slim terms have been run systematically on NetGrep. Results are reported for those schemas with at least 5 but no more than 80,000 instances in *S. cerevisiae*: 780 triangular schemas; 80,719 4-node linear schemas; and 30,642 4-node branched schemas. Boxplots of the running times for each topology are given; boxplots are a convenient way of depicting the smallest observation, second quartile, median, third quartile, and largest observation in the data.

schemas, almost all have fewer than 80,000 instances in *S. cerevisiae* (all triangular schemas, 97,170 quad schemas and 37,129 Y-star schemas). Statistics about how long NetGrep takes to retrieve all instances for each query that has between 5 and 80,000 instances in yeast are given in Figure 3.3; we exclude schemas with fewer than 5 matches as they typically take less time. As can be seen, matches for each of these queries are found within 100 seconds, but the vast majority in fact take less than even 10 seconds. We are not able to time NetMatch and Narada in a systematic manner; thus, we have arbitrarily chosen three triangle, five quad, and five Y-star molecular function queries, to give a sampling of run times for these previous approaches on these types of schemas. The schemas and their timings are shown in Table 3.3.

Topology	Query	Narada	Net Match	Net Grep
Triangle	GO:0003677, GO:0004386, GO:0004672		15	0.1
Triangle	GO:0004386, GO:0004672, GO:0030528		16	0.2
Triangle	GO:0003723, GO:0003723, GO:0003723		15	1.9
Quad	GO:0004386, GO:0003677, GO:0016874, GO:0016829	1	14	0.2
Quad	GO:0016787, GO:0030234, GO:0005515, GO:0008233	2.3	17	1.2
Quad	GO:0003677, GO:0003723, GO:0005515, GO:0005198	4	16	1.9
Quad	GO:0016787, GO:0005198, GO:0003677, GO:0016779	2.2	17	1.7
Quad	GO:0016787, GO:0016740, GO:0016779, GO:0030528	4.8	16	2.9
Y-star	GO:0008233, GO:0016874, GO:0030234, GO:0005215		15	0.2
Y-star	GO:0005515, GO:0004721, GO:0008233, GO:0016740		17	0.8
Y-star	GO:0005515, GO:0008233, GO:0005198, GO:0005215		17	3.9
Y-star	GO:0030528, GO:0005515, GO:0016740, GO:0005215		14	1.5
Y-star	GO:0016740, GO:0005515, GO:0030528, GO:0005215		14	5.2

Table 3.3: A comparison of running times (in seconds) for several sample schemas annotated with GO molecular function slim terms on the *S. cerevisiae* interaction network using Narada, NetMatch and NetGrep. Of the previous methods, Narada and NetMatch are chosen as they can be run off-the-shelf for these schemas; note, however, that Narada only handles linear topology queries. All reported running times are for search and output only. In the case of the Y-stars, the first term shown annotates the central node. The schemas shown have between 10 and 11,000 instances in *S. cerevisiae*.

3.5 Conclusions

In this chapter, we have described fast algorithms for performing general network schema searches within biological networks. These algorithms are the heart of our system, NetGrep, which allows a wider range of network queries than possible with related approaches. In the cases where direct comparisons with other tools can be made, NetGrep consistently finds matches much more quickly.

Users are not restricted to the node features or edge types in the data files provided with NetGrep. All features are easily expandable to include virtually any description or type. In fact, users are not confined to biological networks at all: NetGrep can incorporate any data which can be represented in network format, which allows network schema analysis to be extended to many other areas of research. Using Net-

Grep to analyze network schemas in social networks is just one of many avenues for interesting future work.

Most interaction networks provide only a static view of the interactome; that is, they describe which proteins interact with one another but do not tell anything about when the interactions occur. Therefore, the network schemas found in such an interaction network will also by definition be static. By incorporating appropriate gene expression data into the interaction networks and then ensuring that all edges in a query schema have the coexpression type (ANDed with any other desired edge type), the results returned by NetGrep will be dynamic and more accurately reflect the true organizational patterns of the cell. A recent paper found that cells use different timing activity motifs, which capture patterns in the dynamic use of a network [8]; perhaps NetGrep could be used to help find such motifs in interactomes.

Chapter 4

Conclusion

In this thesis, we have introduced the powerful concept of network schemas, labeled subgraphs that incorporate both the attributes of proteins and the topology in which they interact with one another. In a large-scale analysis, we have shown that network schemas describe organizational units within interactomes. In chapter 2, we presented a fully-automated procedure for discovering network schemas along with an analysis of schemas which we found in *S. cerevisiae* and *H. sapiens*. In chapter 3, we introduced NetGrep, a powerful system for searching protein interactomes for instances of a diverse set of user-supplied network schemas.

There are many avenues for extending our current work on network schemas. For our framework for automatically inferring emergent network schemas, we have examined four of the most basic topologies for schemas; however, additional topologies may also be considered. Also, by allowing certain proteins in a schema to be optional, we may find that the added flexibility enables us to find yet more emergent schemas. When possible, taking the contextual information about interactions into consideration would make our procedure for determining emergent schemas more accurate. Extending our work to networks with more than just physical-physical protein interactions should provide interesting results. A comparison and analysis of emer-

gent schemas occurring in interactomes across the evolutionary spectrum could shed much light on how networks expand or change to incorporate new motifs or protein functions.

Our network schema search system, NetGrep, allows a wide-range of possible queries that supercede many previously studied interaction patterns. However, we predict that as the tool becomes more widely used, additional features will be requested and required by users. Furthermore, while the algorithm we described for solving the labeled subgraph isomorphism problem is fast and effective in practice for biological networks, as we scale up and allow the user more flexibility— for example, by permitting schemas with more matches or by enabling the user to batch searches—more sophisticated algorithms may be necessary. In some cases, an approximate approach may be necessary.

This thesis has taken an important first step towards defining a computational methodology for analyzing biological networks. Since large-scale protein interaction networks are being determined at an increasing pace, we anticipate that network schema analysis, as introduced here, will become an increasingly important means for determining how proteins work together in the cell.

Bibliography

[1] Alon, N., Dao, P., Hajirasouliha, I., Hormozdiari, F., and Sahinalp, S. Biomolecular network motif counting and discovery by color coding. *Bioinformatics 24* (2008), i241–i249.

[2] Ashburner, M., Ball, C., Blake, J., Botstein, D., Butler, H., Cherry, J., et al. Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet. 25*, 1 (2000), 25–29.

[3] Barabasi, A., and Oltvai, Z. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics 5* (2004), 101–103.

[4] Bateman, A., Coin, L., Durbin, R., Finn, R., Hollich, V., and Griffiths-Jones, S. The Pfam protein familes database. *Nucleic Acids Res. 32* (2004), D138–D141.

[5] Bornberg-Bauer, E., Beaussart, F., Kummerfeld, S., Teichmann, S., and 3rd, J. W. The evolution of domain arrangements in proteins and interaction networks. *Cell Mol Life Sci 62* (2005), 435–45.

[6] Boyle, E. I., Weng, S., Gollub, J., Jin, H., Botstein, D., Cherry, J. M., and Sherlock, G. GO::TermFinder–open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics 20*, 18 (2004), 3710–3715.

[7] Breitkreutz, B., Stark, C., and Tyers, M. Osprey: a network visualization system. *Genome Biology 4* (2003), R22.

[8] Chechik, G., Oh, E., Rando, O., Weissman, J., Regev, A., and Koller, D. Activity motifs reveal principles of timing in transcriptional control of the yeast metabolic network. *Nature Biotechnology 26* (2008), 1251–1259.

[9] Chomez, P., Backer, O. D., Bertrand, M., Plaen, E. D., Boon, T., and Lucas, S. An overview of the MAGE gene family with the identification of all human members of the family. *Cancer Res 61*, 14 (2001), 5544–5551.

[10] Cook, D., and Holder, L. Graph-based data mining. *IEEE Intelligent Systems 15*, 2 (2000), 32–41.

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

[12] Despons, L., Wirth, B., Louis, V., Potier, S., and Souciet, J.-L. An evolutionary scenario for one of the largest yeast gene families. *Trends in Genetics 22* (2006), 10–15.

[13] Dost, B., Shlomi, T., Gupta, N., Ruppin, E., Bafna, V., and Sharan, R. Qnet: A tool for querying protein interaction networks. In *RECOMB* (2007), T. P. Speed and H. Huang, Eds., vol. 4453, Springer, pp. 1–15.

[14] Ferro, A., Giugno, R., Pigola, G., Pulvirenti, A., Skripin, D., Bader, G., and Sasha, D. NetMatch: a Cytoscape plugin for searching biological networks. *Bioinformatics 23* (2007), 910–912.

[15] Flannick, J., Novak, A., Srinivasan, B., McAdams, H., and Batzoglou, S. Graemlin: general and robust alignment of multiple large interaction networks. *Genome Res. 16*, 9 (2006), 1169–81.

[16] Fong, J., Geer, L., Panchenko, A., and Bryant, S. Modeling the evolution of protein domain architectures using maximum parsimony. *J. Mol. Biol. 366* (207), 307–315.

[17] Ghazizadeh, S., and Chawathe, S. SEuS: Structure extraction using summaries. In *Proceedings of the 5th International Conference on Discovery Science* (2002), pp. 71–85.

[18] Giot, L., Bader, J., Brouwer, C., Chaudhuri, A., Kuang, B., Li, . Y., et al. A protein interaction map of Drosophila melanogaster. *Science 302* (2003), 1727–1736.

[19] Giugno, R., and Shasha, D. Graphgrep: A fast and universal method for querying graphs. In *Proc Int Conf Pattern Recognit (ICPR), Quebec, Canada* (2002), pp. 112–115.

[20] Gomez, S., Lo, S.-H., and Rzhetsky, A. Probabilistic prediction of unknown metabolic and signal-transduction networks. *Genetics 159* (2001), 1291–1298.

[21] Gough, J., Karplus, K., Hughey, R., and Chothia, C. Assignment of homology to genome sequences using a library of hidden Markov models that represent all proteins of known structure. *J Mol Biol 313* (2001), 903–919.

[22] Grochow, J., and Kellis, M. Network motif discovery using subgraph enumeration and symmetry breaking. In *Lecture Notes in Bioinformatics* (2007), vol. 4453, New York; Springer-Verlag, pp. 92–106.

[23] Guimaraes, K., Jothi, R., Zotenko, E., and Przytycka, T. Predicting domain-domain interactions using a parsimony approach. *Genome Biology 7* (2006), R104.

[24] Han, J.-D., Bertin, N., Hao, T., Goldberg, D., Berriz, G., Zhang, L., et al. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature 430* (2004), 88–93.

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

[26] Harris, M., Clark, J., Ireland, A., Lomax, J., Ashburner, M., Foulger, R., Eilbeck, K., Lewis, S., Marshall, B., Mungall, C., Richter, J., Rubin, G., Blake, J., Bult, C., Dolan, M., Drabkin, H., Eppig, J., Hill, D., Ni, L., Ringwald, M., Balakrishnan, R., Cherry, J., Christie, K., Costanzo, M., Dwight, S., Engel, S., Fisk, D., Hirschman, J., Hong, E., Nash, R., et al. The gene ontology (GO) database and informatics resource. *Nucleic Acids Res 32* (2004), D258–D261.

[27] Hartwell, L., Hopfield, J., Leibler, S., and Murray, A. From molecular to modular cell biology. *Nature 402* (1999), C47–52.

[28] He, W., and Parker, R. Functions of Lsm proteins in mRNA degradation and splicing. *Current Opinion in Cell Biology 12* (2000), 346–350.

[29] Hong, E., Balakrishnan, R., Christie, K., Costanzo, M., Dwight, S., Engel, S., et al. Saccharomyces Genome Database. http://www.yeastgenome.org.

[30] Huan, J., Wang, W., and Prins, J. Efficient mining of frequent subgraphs in the presence of isomorphism. In *Proceedings of the Third IEEE International Conference on Data Mining* (Washington, DC, USA, 2003), IEEE Computer Society, p. 549.

[31] Huan, J., Wang, W., Prins, J., and Yang, J. SPIN: mining maximal frequent subgraphs from graph databases. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2004), ACM, pp. 581–586.

[32] Hulo, N., Sigrist, C., Saux, V. L., Langendijk-Genevaux, P., Bordoli, L., Gattiker, A., de Castro, E., Bucher, P., and Bairoch, A. Recent improvements to the PROSITE database. *Nucleic Acids Res 32* (2004), D134–D137.

[33] Inokuchi, A., Washio, T., and Motoda, H. An apriori-based algorithm for mining frequent substructures from graph data. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery* (London, UK, 2000), Springer-Verlag, pp. 13–23.

[34] Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., and Sakaki, Y. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci. USA. 98*, 8 (2001), 4569–4574.

[35] Itzhaki, Z., Akiva, E., Altuvia, Y., and Margalit, H. Evolutionary conservation of domain-domain interactions. *Genome Biology 7* (2006), R125.

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

[37] Kelley, B., Sharan, R., Karp, R., Sittler, T., Root, D., Stockwell, B., and Ideker, T. Conserved pathways within bacteria and yeast as revealed

by global protein network alignment. *Proc. Natl. Acad. Sci. USA 100* (2003), 11394–11399.

[38] Kelley, B., Yuan, B., Lewitter, F., Sharan, R., Stockwell, B., and Ideker, T. Pathblast: a tool for alignment of protein interaction networks. *Nucleic Acids Res 32* (2004), 83–88.

[39] Kiemer, L., Costa, S., Ueffing, M., and Cesareni, G. WH-PHI: A weighted yeast interactome enriched for direct physical interactions. *Proteomics 7* (2007), 932–943.

[40] Kim, P., Lu, L., Xia, Y., and Gerstein, M. Relating three-dimensional structures to protein networks provides evolutionary insights. *Science 314* (2006), 1938–1941.

[41] Kim, P., Sboner, A., Xia, Y., and Gerstein, M. The role of disorder in interaction networks: a structural analysis. *Molecular Systems Biology 4* (2008), 179.

[42] Koyuturk, M., Kim, Y., Topkara, U., Subramaniam, S., Szpankowski, W., and Grama, A. Pairwise alignment of protein interaction networks. *J Comput Biol 13* (2005), 182–199.

[43] Kuramochi, M., and Karypis, G. Frequent subgraph discovery. In *Proceedings of the 1st IEEE Conference on Data Mining* (2001), pp. 313–320.

[44] Kuramochi, M., and Karypis, G. Finding frequent patterns in a large sparse graph. *Data mining and knowledge discovery 11* (2005), 243–271.

[45] Lacroix, V., Fernandes, C., and Sagot, M.-F. Motif search in graphs: Application to metabolic networks. *IEEE Transactions on computational biology and bioinformatics 3*, 4 (2006), 360–368.

[46] Lee, T., Rinaldi, N., Robert, F., Odom, D., Bar-Joseph, Z., Gerber, G., et al. Transcriptional regulatory networks in Saccharomyces cerevisiae. *Science 298* (2002), 799–804.

[47] Letunic, I., Copley, R., Schmidt, S., Ciccarelli, F., Doerks, T., Schultz, J., Ponting, C., and Bork, P. SMART 4.0: towards genomic data integration. *Nucleic Acids Res 32* (2004), D142–D144.

[48] Luscombe, N., Babu, M., Yu, H., Snyder, M., Teichmann, S., and Gerstein, M. Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature 431* (2004), 308–312.

[49] Maslov, S., and Sneppen, K. Specificity and Stability in Topology of Protein Networks. *Science 296*, 5569 (2002), 910–913.

[50] Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. Network motifs: simple building blocks of complex networks. *Science 298* (2002), 824–827.

[51] Mitsui, K., Fumihiro, O., Nakamura, N., Yoshihide, D., Hiroki, I., and Kanazawa, H. A novel membrane protein capable of binding the Na^+/H^+ antiporter (Nha1p) enhances the salinity-resistant cell growth of Saccharomyces cerevisiae. *J. Biol. Chem. 279* (2004), 12438–47.

[52] Nabieva, E., Jim, K., Agarwal, A., Chazelle, B., and Singh, M. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics 21 Suppl. 1* (2005), i302–i310.

[53] Neduva, V., Linding, R., Su-Angrand, I., Stark, A., de Masi, F., Gibson, T., et al. Systematic discovery of new recognition peptides mediating protein interaction networks. *PLOS Biol. 3*, 12 (2005), e405.

[54] Nye, T., Berzuini, C., Gilks, W., Babu, M., and Teichmann, S. Statistical analysis of domains in interacting protein pairs. *Bioinformatics 21* (2005), 993–1001.

[55] Pagel, P., Wong, P., and Frishman, D. A domain interaction map based on phylogenetic profiling. *Journal of Molecular Biology 5* (2004), 1331–1346.

[56] Pandey, J., Koyuturk, M., Kim, Y., Szpankowski, W., Subramanian, S., and Grama, A. Functional annotation of regulatory pathways. *Bioinformatics 23* (2007), i377–i386.

[57] Pao, S., Paulsen, I., and Saier, M. Major facilitator superfamily. *Microbiology and Molecular Biology Reviews 62* (1998), 1–34.

[58] Patton, E., Willems, A., and Tyers, M. Combinatorial control in ubiquitin-dependent proteolysis: don't Skip the F-box hypothesis. *Trends in Genetics 14* (1998), 236–243.

[59] Pawson, T., and Nash, P. Assembly of cell regulatory systems through protein interactions. *Science 300* (2003), 445–452.

[60] Pinter, R., Rokhlenko, O., Yeger-Lotem, E., and Ziv-Ukelson, M. Alignment of metabolic pathways. *Bioinformatics 21* (2005), 3401–3408.

[61] Przytycka, T., Davis, G., Song, N., and Durand, D. Graph theoretical insights into evolution of multidomain proteins. *J. of Comp. Biol. 13* (2006), 351–363.

[62] Ptacek, J., Devgan, G., Michaud, G., Zhu, H., Zhu, X., Fasolo, J., et al. Global analysis of protein phosphorylation in yeast. *Nature 438* (2005), 679–684.

[63] Riley, R., Lee, C., Sabatti, C., and Eisenberg, D. Inferring protein domain interactions from databases of interacting proteins. *Genome Biology 6* (2005), R89.

[64] Rives, A., and Galitski, T. Modular organization of cellular networks. *Proc. Natl. Acad. Sci. USA 100*, 3 (2003), 1128–1133.

[65] Sandmann, T., Herrmann, J. M., Dengjel, J., Schwarz, H., and Spang, A. Suppression of coatomer mutants by a new protein family with COPI and COPII binding motifs in Saccharomyces cerevisiae. *Mol. Biol. Cell 14*, 8 (2003), 3097–3113.

[66] Schreiber, F., and Schwobbermeyer, H. Mavisto: a tool for the exploration of network motifs. *Bioinformatics 21* (2005), 3572–3574.

[67] Schultz, J., Milpetz, F., Bork, P., and Ponting, C. SMART, a simple modular architecture research tool: Identification of signaling domains. *Proc Natl Acad Sci U S A 95* (1998), 5857–5864.

[68] Scott, J., Ideker, T., Karp, R., and Sharan, R. Efficient algorithms for detecting signaling pathways in interaction networks. *Journal of Computational Biology 13*, 2 (2006), 133–144.

[69] Shannon, P., Markiel, A., Ozier, O., Baliga, N., Wang, J., Ramage, D., Amin, N., Schwikowski, B., and Ideker, T. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Res. 13* (2003), 2498–2504.

[70] Sharan, R., and Ideker, T. Modeling cellular machinery through biological network comparison. *Nature Biotechnology 24* (2006), 427–433.

[71] Sharan, R., Suthram, S., Kelley, R., Kuhn, T., McCuine, S., Uetz, P., et al. Conserved patterns of protein interaction in multiple species. *Proc. Natl Acad. Sci. USA 102* (2005), 1974–1979.

[72] Sharan, R., Ulitsky, I., and Shamir, R. Network-based prediction of protein function. *Molecular Systems Biology 3* (2007), 88.

[73] Shen-Orr, S., Milo, R., Mangan, S., and Alon, U. Network motifs in the transcriptional regulation network of E. coli. *Nat. Genet. 31* (2002), 64–68.

[74] Shlomi, T., Segal, D., Ruppin, E., and Sharan, R. QPath: a method for querying pathways in a protein-protein interaction network. *BMC Bionformatics 7* (2006), 199.

[75] Singh, R., Xu, J., and Berger, B. Pairwise global alignment of protein interaction networks by matching neighborhood topology. In *Proceedings of the 11th International Conference on Research in Comp. Mol. Biol (RECOMB)* (2007), pp. 16–31.

[76] Sivars, U., Aivazian, D., and Pfeffer, S. R. Yip3 catalyses the dissociation of endosomal Rab-GDI complexes. *Nature 425* (2003), 856–859.

[77] Spirin, V., and Mirny, L. A. Protein complexes and functional modules in molecular networks. *Proc. Natl. Acad. Sci. USA. 100* (2003), 12123–12128.

[78] Sprinzak, E., and Margalit, H. Correlated sequence-signatures as markers of protein-protein interaction. *J. Mol. Biol. 311* (2001), 681–692.

[79] Sprinzak, E., Sattath, S., and Margalit, H. How reliable are experimental protein-protein interaction data? *J. Mol. Biol. 327*, 5 (2003), 919–923.

[80] Stark, C., Breitkreutz, B., Reguly, T., Boucher, L., Breitkreutz, A., and Tyers, M. BioGRID: A general repository for interaction datasets. *Nucleic Acids Res. 34* (2006), D535–D539.

[81] Steffen, M., Petti, A., Aach, J., D'haeseleer, P., and Church, G. Automated modelling of signal transduction networks. *BMC Bionformatics 3* (2002), 34.

[82] Stuart, J. M., Segal, E., Koller, D., and Kim, S. K. A gene-coexpression network for global discovery of conserved genetic modules. *Science 302* (2003), 249–255.

[83] Tian, Y., McEachin, R., Santos, C., States, D., and Patel, J. Saga: a subgraph matching tool for biological graphs. *Bioinformatics 23*, 2 (2007), 232–239.

[84] Tong, A., Drees, B., Nardelli, G., Bader, G., Brannetti, B., Castagnoli, L., et al. A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. *Science 295* (2002), 321–324.

[85] Tong, A., Lesage, G., Bader, G., Ding, H., Xu, H., Xin, X., Young, J., Berriz, G., Brost, R., Chang, M., Chen, Y., Cheng, X., Chua, G., Friesen, H., Goldberg, D., Haynes, J., Humphries, C., He, G., Hussein, S., Ke, L., Krogan, N., Li, Z., Levinson, J., Lu, H., Minard, P., Munyana, C., Parsons, A., Ryan, O., Tonikian, R., Roberts, T., et al. Global mapping of the yeast genetic interaction network. *Science 303* (2004), 808–813.

[86] von Mering, C., Huynen, M., Jaeggi, D., Schmidt, S., Bork, P., and Snel, B. STRING: a database of predicted functional associations between proteins. *Nucleic Acids Res 31* (2003), 258–261.

[87] Wernicke, S., and Rasche, F. Fanmod: a tool for fast network motif detection. *Bioinformatics 22* (2006), 1152–1153.

[88] Wojcik, J., and Schacter, V. Protein-protein interaction map inference using interacting domain profile pairs. *Bioinformatics 17 Suppl 1* (2001), S296–S305.

[89] Wong, S., Zhang, L., Tong, A., Li, Z., Goldberg, D., King, O., Lesage, G., Vidal, M., Andrews, B., Bussey, H., Boone, C., and Roth, F. Combining biological networks to predict genetic interactions. *Proc Natl Acad Sci U S A 101* (2004), 15682–15687.

[90] Wuchty, S., Oltvai, Z., and Barabasi, A. Evolutionary conservation of motif constituents in the yeast protein interaction network. *Nat. Genet. 35* (2003), 176–179.

[91] Yang, X., Matern, H. T., and Gallwitz, D. Specific binding to a novel and essential Golgi membrane protein (Yip1p) functionally links the transport GTPases Ypt1p and Ypt31p. *The EMBO Journal 17* (1998), 4954–4963.

[92] Yeger-Lotem, E., Sattath, S., Kashtan, N., Izkovitz, S., Milo, R., Alon, U., and Margalit, H. Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction. *Proc. Natl. Acad. Sci. USA 101*, 16 (2004), 5934–5939.

[93] Zhang, L., King, O., Wong, S., Goldberg, D., Tong, A., Lesage, G., Andrews, B., et al. Motifs, themes and thematic maps of an integrated Saccharomyces cerevisiae interaction network. *J. Biol 4* (2005), 6.

[94] Zhu, X., Gerstein, M., and Snyder, M. Getting connected: analysis and principles of biological networks. *Genes and Development 21* (2007), 1010–1024.