On the Structure and Evolution of Protein Interaction Networks

by

Joshua A. Grochow

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

Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 2006

Author .....
Department of Electrical Engineering and Computer Science
Aug 11, 2006

Certified by .....
Manolis Kellis
Assistant Professor
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

On the Structure and Evolution of Protein Interaction Networks

by

Joshua A. Grochow

Submitted to the Department of Electrical Engineering and Computer Science
on Aug 11, 2006, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

The study of protein interactions from the networks point of view has yielded new insights into systems biology [Bar03, MA03, RSM+02, WS98]. In particular, "network motifs" become apparent as a useful and systematic tool for describing and exploring networks [BP06, MKFV06, MSOI+02, SOMMA02, SV06]. Finding motifs has involved either exact counting (e.g. [MSOI+02]) or subgraph sampling (e.g. [BP06, KIMA04a, MZW05]). In this thesis we develop an algorithm to count all instances of a particular subgraph, which can be used to query whether a given subgraph is a significant motif. This method can be used to perform exact counting of network motifs faster and with less memory than previous methods, and can also be combined with subgraph sampling to find larger motifs than ever before – we have found motifs with up to 15 nodes and explored subgraphs up to 20 nodes. Unlike previous methods, this method can also be used to explore motif clustering and can be combined with network alignment techniques [FNS+06, KSK+03].

We also present new methods of estimating parameters for models of biological network growth, and present a new model based on these parameters and underlying binding domains.

Finally, we propose an experiment to explore the effect of the whole genome duplication [KBL04] on the protein-protein interaction network of *S. cerevisiae*, allowing us to distinguish between cases of subfunctionalization and neofunctionalization.

Thesis Supervisor: Manolis Kellis
Title: Assistant Professor

Acknowledgments

In general, I would like to acknowledge the members of the Kellis Lab for Computational Biology at M.I.T. during the 2005-2006 academic year and the summer of 2006. In particular, I would like to thank Doctor Hui Ge of the Whitehead Institute, Pouya Kheradpour, Mike Lin, Patrycja Missiuro, Aviva Presser, Matt Rasmussen, Alex Stark, and Radek Szklarczyk for many interesting and humorous conversations, and for their useful suggestions and insights into both the content and the presentation of this thesis. I would like to thank Nikki Pfarr for working so hard on Fig. 3-8 to get it just right. I would also like to thank Professor Manolis Kellis for all his assistance and guidance. Finally, I would like to thank my family and friends for all their patience and support during this incredibly busy time. This work was supported in part by startup funds from Professor Kellis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Network Science

Much of the world consists of complex systems. This complexity stems from chaotic dynamics and emergent behaviors, or because of myriads of smaller units interacting in complicated ways, or combinations of both phenomena. The study of complex systems is an important stepping stone towards what E. O. Wilson calls "consilience" [Wil98] – the unification of the sciences, the social sciences, and the arts.

Network science is the study of complex systems (of interacting parts) from a graph-theoretic point of view. Network science has yielded many insights into the way our world works, from the cell, to the internet, to the economy. Network science has its roots in the study of social networks as early as the 1950s, but in the past decade or so it has seen an explosion of activity, due both to the growing availability of network data – viz. the World Wide Web [Bar03], protein interaction networks [HBH+04], and food webs [CBN90, WM00] (see §2.1 for more examples) – and also the availability of personal computers powerful enough to crunch on that data.

Today network science brings together researchers from nearly all branches of academia and instury, who have realized a common underlying theme in their work: "It's the network, stupid!" Network science has entered the public consciousness through debates on using social network analysis to catch terrorists, and through the popular literature such as Albert-László Barabási's *Linked* [Bar02], Malcolm Glad-

well's *The Tipping Point* [Gla02], Steven Strogatz's *Sync* [Str03], and Duncan Watts' *Six Degrees* [Wat03].

Although this thesis focuses on biological networks, and in particular the protein-protein interaction network of *Saccharomyces cerevisiae* (baker's yeast), many of the methods and ideas herein – particularly those in Chapter 2 (background material) and Chapter 3 (a new approach to discovering network motifs) – are applicable to any network, and thus to many different realms of science and social science. Moreover, the research on network evolution in Chapters 4 and 5 may inspire the design of complex systems engineered for extensibility and evolvability (e.g. [BdW05]).

1.2 Network Motifs

Modularity has been standard practice in systems design and engineering for decades. Modular structure enables the re-use of common sub-parts. Engineers often impose hierarchical organization to larger systems in order to help manage and control their complexity. In addition, network science has also found that these properties are prevalent in naturally occurring, evolving, and growing networks [RSM+02, HBH+04, MSOI+02]. Studying these naturally occurring sub-networks has yielded insights into the information-processing roles of sets of nodes in a network [MA03, SOMMA02].

Network motifs provide an important viewpoint for understanding the modularity and the overall structure of networks [KMP+01, MZA03, RRSA02, ZMR+04]. Motifs were first introduced in [MSOI+02]. The importance of network motifs as information-processing modules was modeled theoretically in [SOMMA02] and [MA03], and verified experimentally in [KMP+01], [MZA03], [RRSA02], and [ZMR+04].

Network motifs are defined as subnetworks that are significant or non-random in one or more ways [MSOI+02]. Network motifs are typically determined by over-representation compared to randomized versions of a network [BP06, MSOI+02, SOMMA02]. Similarly, antimotifs are determined by underrepresentation compared to randomized versions of a network. However, recent work in aligning biological networks [FNS+06, KSK+03] reveals conserved sub-networks, and we think of these

conserved sub-networks as a new type of network motif. The former, more standard type of motif may be distinguished as frequency-based, and the latter as conservation-based motifs. Unless otherwise specified, this thesis will always refer to frequency-based motifs, which have also found applications in other areas of network science.

Two basic methodologies are available for finding network motifs: exact counting (e.g. [MSOI$^+$02]) or subgraph sampling (e.g. [BP06, KIMA04a, MZW05]). Both methods attempt determine the significance of all (or many, in the case of sampling) subgraphs of a given size by comparing their frequency to their frequency in a random ensemble of networks. It is generally thought that a qualitative description of the significance of a particular subgraph (e.g. whether or not it is a motif) is more informative than a quantitative one (e.g. its z-score). To determine which graphs are motifs or antimotifs, subgraph sampling [BP06, KIMA04a, MZW05] is effective and efficient, and can determine the significance of larger subgraphs than the current methods of exact counting.

However, it is necessary to find all instances of a given graph as subgraphs of a network to

(a) determine whether a given graph (perhaps determined experimentally) is a significant motif,

(b) explore motif clustering to see how motifs may be parts of larger structures and to see how dependent a motif's significance is upon the accuracy of the network, and

(c) combine frequency-based motif finding with conservation-based motif finding (network alignment) in certain ways.

In Chapter 3 we present a method to achieve this. By analogy with sequence motifs, we believe combining frequency-based motif finding with conservation-based motif finding will be a particularly important application of this new approach to finding network motifs. In addition to these applications, the method can also be used to get the exact counts of all subgraphs of a given size, and it does so faster than previous methods.

1.3 Evolution of Biological Networks

Cellular processes are defined by the way proteins interact with one another, with the environment, and with DNA. Understanding how proteins interact with one another at the chemical level is an important first step to modelling cellular processes, and can reveal principles important in capturing the larger systems picture revealed by protein-protein interaction networks and gene regulatory networks.

In a protein-protein interaction (PPI) network, the nodes are proteins and the (undirected) edges represent pairwise protein-protein interactions. In a gene regulatory network, nodes represent genes and proteins, and (directed) edges represent the production of a protein by a gene or the regulation of a gene by a protein.

In the few short years they have been around, PPI nets have already found several applications. They have been used to predict domain-domain interactions [DMSC02], to predict *de novo* protein-peptide interactions based on network motifs [RS04], and to annotate previously unclassified genes by correlating a PPI net with a protein-DNA net [MBV03].

In this thesis, we focus on the PPI net of the yeast *Saccharomyces cerevisiae* for three reasons: *S. cerevisiae* is possibly one of the most well-studied, simple organisms on the planet (along with *E. coli*; *H. sapiens* is well-studied, but much more complex), the complete genomes of *S. cerevisiae* and 11 of its close relatives are available, and *S. cerevisiae* has a rich evoluationary history, including in particular a whole genome duplication (WGD) [KBL04].

Genetic duplication, whether at the scale of a single gene, a chromosomal segment, or a whole genome, is a significant mechanism in evolution [Ohn70]. When duplication occurs, it is believed that the two members of a duplicate pair initially have identical functions and interactions. Afterwards, there is a transient period during which one of the duplicates will differentiate, diverge in function, or disappear altogether [Wag02]. If the two duplicates differentiate so as to each take on a different part of the ancestral function, they are said to have undergone **subfunctionalization**. If one of the duplicates stays the same and the other takes on a novel function, it is said

to have undergone **neofunctionalization**. Exactly how the divergence of duplicates occurs, and the relative importance of these two processes, are still open questions.

In Chapters 4 and 5, we explore the effects of the WGD on the evolution of the yeast PPI net. Chapter 4 introduces a model of network growth that reproduces many properties of the actual yeast PPI net, and treats WGD and single-gene duplication in a unified framework. In Chapter 5 we propose an experiment to study the divergence of whole genome duplicates, and specifically to examine the asymmetry of divergence. These experiments will also distinguish between cases of subfunctionalization and neofunctionalization.

1.4 Organization

The remainder of this thesis is organized as follows. Chapter 2 goes over introductory background material, which introduces terminology, definitions, and fundamental issues in networks generally and biological networks specifically. Chapter 3 introduces a new approach for detecting network motifs, which is faster than previous methods, and also applicable to many other tasks that are wholly unavailable to previous methods. Whereas previous methods have only been able to discover motifs up to 8 nodes due to combinatorial scaling, we present a motif of 15 nodes, and explore subgraphs of 10 and 20 nodes with our new method, along with some basic analysis showing that these larger motifs represent biologically relevant structures. Chapter 4 (joint work with Alexei Vázquez, Matt Rasmussen, Manolis Kellis, and Albert-László Barabási) introduces new methods for estimating parameters of network growth, providing a biological grounding to models that were previously solely theoretical. Chapter 5 (in collaboration with Jean-François Rual and Marc Vidal) discusses asymmetric divergence of duplicated genes, and proposes experiments that will allow us to explore this divergence and distinguish between instances of subfunctionalization and neofunctionalization. More details may be found in Appendix C. The experiments are being carried out in the lab of Marc Vidal at the Dana Farber Cancer Institute of Harvard, and will be completed after the submission of this thesis.

Chapter 2

Background

In this chapter we begin with examples of real-world networks (as opposed to theoretical graphs) (§2.1), and we review some basic terminology associated to networks and graphs (§2.2). We then review several interesting properties discovered in analyzing real-world networks (§2.3), and recent work that unifies them with network motifs, and provides a systematic framework for the analysis of networks (§2.4). Finally, we review network motifs (§2.5), and some of the basic techniques and ideas that will be used in Chapter 3, including the limitations of current methods of discovering network motifs.

2.1 Networks in the Real World

Many real world networks have been studied in the past decade (see Table 2.1). It is important to note that nearly all of these networks rely on raw data that is subject to both experimental and human error, and sampling or ascertainment bias. For example, the protein-protein interaction network of the yeast *Saccharomyces cerevisiae* has been probed by several types of experiments, and each type of experiment has been performed independently by several groups. This produces several networks with different biases, that often conflict with one another, and are probably still incomplete. Several studies have been done on the effects of experimental bias [KGG06] and on the effects of combining different network datasets [GSW04, HRO05, HV03]. Because

of these potential inaccuracies, global statistics must be robust in order to examine the properties of actual networks based on experimental data.

Network	Nodes	Edges	Directed?
Internet	routers	physical connections	undirected
WWW	web pages	hyperlinks	directed
Social networks	people	social ties	mixed
Organizational networks	people	reporting / directing	directed
Sexual networks	people	sexual relations	undirected
Food webs	species	predator-prey relations	directed
Protein interaction networks	proteins	protein interactions	mixed
Metabolic networks	substrates and enzymes	metabolic pathways	directed
Genetic regulatory networks	genes or gene products	regulation of gene expression	directed

Table 2.1: Real-world networks.

2.2 Network/Graph Terminology

Graphs and networks. The terms **graph** and **network** are used interchangeably (though we typically use "graph" to refer to smaller graphs, such as motifs, and "network" to refer to the real-world networks being studied). A graph is a set of **vertices** or **nodes**, which may be connected in pairs by **edges**. Typically, graphs are denoted by the capital letters $G, H, \ldots$, vertices are denoted by the lowercase letters $u, v, w, \ldots$, and edges are denoted by the lowercase letters $e, f, \ldots$. The set of vertices is often denoted V or $V(G)$ if the implied graph is not clear from context. Similarly, the set of edges is denoted by E or $E(G)$.

Subgraphs. $H \subset G$ is used to denote that H is a **subgraph** of G, i.e. that $V(H) \subset V(G)$ and $E(H) \subset E(G)$ such that both endpoints u, v of each edge in $E(H)$ are present in $V(H)$. A **vertex-based subgraph** (also called an "induced subgraph" or "full subgraph" in graph theory) of a graph G consists of a subset of $V(G)$, and all the edges in G connecting vertices in that subset. To make the

distinction clear, we may refer to subgraphs as "general or edge-based subgraphs". The distinction is particularly relevant to network motifs (§2.5).

Unless otherwise stated, the graphs in this work are undirected and simple: they contain no self-edges, and there can be at most one edge between any pair of vertices. All of the methods in this paper are easily generalizable to graphs with directed edges, and many of the methods are generalizable to graphs with self-edges and multiple-edges.

Random graphs. In 1959, Paul Erdös and Alfréd Rényi introduced the notion of a random graph [ER59]. An Erdős-Rényi random graph on n nodes has an edge between each pair of vertices with fixed probability p. This model of random graph has been well-studied, and it has many attractive properties. For example, the degree distribution of an Erdős-Rényi graph is a Poissonian, sharply peaked around an average degree, and the size of the largest connected component undergoes a well-characterized phase transition as p increases.

In network science, however, Erdős-Rényi random graphs are most often cited as examples of what real networks are *not* (see §2.3). But random graphs in the more general sense still play an important role in the study of real-world networks (see §2.5.2).

2.3 Network Properties and Statistics

Network statistics can be used to discriminate between networks, to evaluate the fidelity of models of network growth, and sometimes to uncover interesting and meaningful properties of a system. In studying networks that grow and evolve – such as the Internet, the World Wide Web, food chains, social networks, and biological networks – a network statistic must be relatively stable with regards to small fluctuations in the network and with regards to measurement inaccuracies. Simple statistics such as number of nodes, number of edges, number of connected components, size of the largest component, and to a certain extent the maximum path length (diameter) provide a rough idea of the structure of the network. In addition, network scientists

have moved beyond these classical graph-theoretic statistics to find more meaningful properties of real-world networks.

For example, for networks that grow and evolve, the relationship between the number of nodes and the number of edges over time can provide interesting insights. This relationship is captured by the **degree distribution**: the number of nodes of each degree. In the classical Erdős-Rényi model of random graphs – in which each pair of nodes is connected by an edge with fixed probability p – the degree distribution is a sharply peaked Poisson distribution around the average degree. The average degree of such networks provides a characteristic scale for the topology of the network.

In many real networks, however, the degree distribution has a very long right tail, and often follows a power law distribution [Bar03]. Such networks are called **scale-free**, because there is no characteristic scale at which interactions take place. Figure 2-1 shows the difference between the distribution in an Erdős-Rényi random graph and a scale-free network.

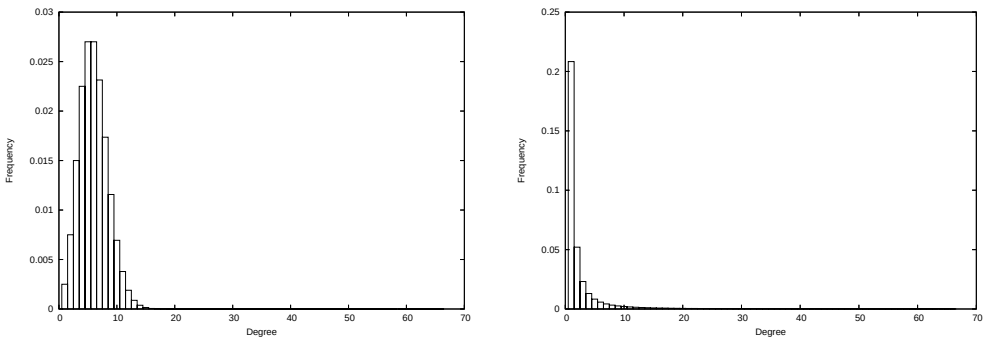


Figure 2-1: The degree distributions of an Erdős-Rényi random graph (left) and a scale-free network (right) with the same average degree.

The exponent α of the power law is sometimes called the "degree exponent". Since $1/k^\alpha$ is never zero, but the graph does not have infinitely many nodes, power law degree distributions always have a cutoff value. Occasionally, this cutoff takes the form of an exponential, such as $P(k) = e^{-k/k_0}\frac{1}{k^\alpha}$. Whereas the average degree was sufficient to characterize the degree distribution of Erdős-Rényi random networks, the degree exponent and the nature of the cutoff are both necessary to characterize

a scale-free distribution.

Recently, questions have been raised as to how well a power law in fact fits the degree distribution of various networks, and also whether or not the exact shape of the degree distribution might be an artifact of the way in which the network data was gathered. Some researchers have thus taken to speaking more generally of "broad-tailed degree distributions" rather than scale-freeness in particular.

More information about the structure of a network can be discerned by examining its **degree correlations**. The degree correlation distribution of a network is the fraction of edges $P(k_1, k_2)$ connecting a node of degree k_1 with a node of degree k_2. A network is **assortative** if the high-degree nodes tend to link to high-degree nodes (i.e. if $P(k_1, k_2)$ is large when k_1 and k_2 are similar), and **disassortative** if high-degree nodes tend to link to low-degree nodes.

Many real networks also exhibit the **small-world** property: they have a large average clustering coefficient and a small diameter [WS98, Wat99]. The **clustering coefficient** of a vertex v is the proportion of pairs of its neighbors that form a triangle with v. If $T(v)$ is the number of triangles intersecting v, then the clustering coefficient $C(v)$ is $T(v)$ divided by $\binom{deg(v)}{2}$.

Real networks also tend to be modular and have a hierarchical structure. The distribution of clustering coefficients has been repoted to capture these properties [RSM+02]. The average clustering coefficient of nodes of degree k is denoted $C(k)$. If $C(k) \sim k^{-\alpha}$, then α is sometimes called the "hierarchical exponent". A network is said to be "hierarchically modular" if $C(k) \sim k^{-1}$ [RSM+02]. It has been shown that models of network growth involving only preferential attachment do not produce hierarchically modular networks, while models incorporating both preferential attachment and node duplication (e.g. genetic duplication) do [Hal04].

Recently, Abdo and de Moura [Ad06] extended the notion of clustering coefficient. The clustering coefficient is the first in a series known as the **clustering profile**. The clustering profile of a vertex v is denoted $C^i(v)$ for $i = 1, 2, 3, \ldots$, where $C^i(v)$ is the proportion of pairs of neighbors of v that are distance i away from one another, where the distance excludes paths that go through v. Thus $C^1(v)$ is the standard clustering

coefficient. Also note that $\sum_i C^i(v) = 1$ (if the sum includes $i = \infty$, i.e. when two neighbors of v are only connected through v). They observe that $\sum_{i=1}^{3} C^i(v)$ is often very close to 1. As an example of the utility of the clustering profile, Abdo and de Moura examine the movie actors network [IMD06], in which two actors are connected if they have acted together in a film. The clustering coefficient $C^1(k)$ decreases rapidly with k for this network, leading one to believe that stars very often work with sets of people that never work with one another. But $C^2(k)$ rises sharply with k, revealing a richer structure.

Many models of network growth have been proposed to explain one or more of these network statistics (with the exception of the clustering profile, because of its recent novelty). Although these statistics are clearly relevant to the structure of a network, it is unclear which properties are most relevant to identifying the process by which a network grew. Network motifs (§2.5) provide yet another property that may capture the finer aspects of a network's structure, and there is a systematic series of properties based on motifs that subsumes both motifs and the other properties listed here ([MKFV06], summarized in §2.4).

2.4 Systematic Network Properties

The properties discussed in §2.3 have shed a great deal of light on the structure of real-world networks. In this section, we present the findings of [VDS+04] and [MKFV06], which provide a systematic set of properties against which models of growth can be evaluated. Furthermore, these systematic properties naturally include many of the properties discussed in §2.3, and by their construction are garaunteed to include any further network properties discovered in the future.

The distribution of subgraphs of a network provides a great deal of information about the network in an unbiased, general framework. The larger the subgraphs counted, the more information the distribution contains. In the limit, the subgraph distribution contains *all* of the information in the network, since every network can be considered its own largest subgraph.

In particular, the clustering coefficient is entirely captured by the distribution of "spoked" subgraphs – subgraphs in which every node is connected to one central node, and perhaps there are other edges. In the case of networks with power law degree and clustering coefficient distributions, even more can be said about the distribution of these "spoked" subgraphs. In [VDS+04], it is shown that the degree exponent γ and the hierarchical exponent α ($P(k) \sim k^{-\gamma}$; $C(k) \sim k^{-\alpha}$) carry the exact same information as the distribution of any *two* spoked subgraphs. In other words, α and γ can be determined by these distributions, and conversely. This can be understood intuitively because the clustering coefficient and the degree distribution are entirely captured by the number of triangles and 3-node lines (sometimes called "wedges") in the graph. Beyond the intuition, this is a very nice analytic result for this particular class of networks, providing further evidence for the above claims.

For this class of power law networks, the spoked subgraphs can be divided into two types [VDS+04]: type I subgraphs, whose number in a random ensemble of networks with fixed (γ, α) follows a power law in the maximal degree k_{max} of the network, and type II subgraphs, whose number remains proportional to the number of nodes in the network. See Table 2.2 for details.

Type	Condition	Distribution
I	$C < 0$	$\sim N k_{max}^{-C}$
II	$C > 0$	$\sim N$

Table 2.2: The two types of subgraphs identified by [VDS+04], defined by their conditions, are distributed as indicated. Here $C = (m - n + 1)\alpha - (n - \gamma)$ where n is the number of nodes in the subgraph, m the number of edges, α the hierarchical exponent of the larger network, and γ the degree exponent.

In [VDS+04] it is also noted that the abundance of subgraphs compared to the total number of nodes leads to subgraph clustering, and the size of these clusters are calculated analytically using methods of percolation theory. More details on motif clustering can be found in §3.4.1.

Subgraph distributions can encompass not only the two parameters in a power law network, but can encompass the exact distributions of nearly all the properties

discussed in §2.3 in networks of any topology, as shown in [MKFV06] and discussed in the remainder of this section.

The distribution of "degree-correlated subgraphs" of d nodes – the so-called dK-series – is exactly the systematic set of network properties desired. A degree-correlated subgraph associates to each node in the subgraph its degree, e.g. a triangle with one node of degree 10, one of degree 3, and one of degree 7. The distribution of degree-correlated subgraphs on one node is exactly the degree distribution. The distribution of degree-correlated subgraphs on two nodes is exactly the degree correlation distribution. The distribution of degree-correlated subgraphs on three nodes captures not only the clustering coefficient (number of triangles), but also the *relationship* between the degree distribution and clustering coefficient distribution.

Note that the symmetries of the subgraphs must be taken into account in these distributions. For example, in the distribution of degree-correlated triangles $P_\Delta(k_1, k_2, k_3)$, the order of the arguments k_1, k_2, k_3 is irrelevant, but in the distribution of degree-correlated lines on 3 nodes, $P_\Lambda(k_1, k_2, k_3)$, the only interchange of arguments that leaves the distribution unchanged is the swapping of k_1 and k_3.

Part of the utility of the dK-series for evaluating models of network growth is that the fidelity of different models can be more easily compared. For example, model A might reproduce the dK-series up to 3 nodes with some error, while model B might reproduce the dK-series up to 4 nodes but with a greater error. Without these properties, such comparisons are all but impossible.

The dK-series can also be used as a very stringent background model for finding network motifs (see §2.5). Milo *et al.* [MSOI$^+$02] introduced the background model whereby the distribution of $(d-1)$-node subgraphs is preserved when identifying d-node motifs. Preserving the dK-series (up to $d-1$ nodes) puts even more information into the background model.

2.5 Network Motifs

Modularity has been standard practice in systems design and engineering for decades. Modular structure enables the re-use of common sub-parts. Engineers often impose hierarchical organization to larger systems in order to help manage and control their complexity. In addition, network science has also found that these properties are prevalent in naturally occurring, evolving, and growing networks [RSM+02, HBH+04, MSOI+02]. Studying these naturally occurring sub-networks has yielded insights into the information-processing roles of sets of nodes in a network [MA03, SOMMA02].

Network motifs provide an important viewpoint for understanding the modularity and the overall structure of networks [KMP+01, MZA03, RRSA02, ZMR+04]. Motifs were first introduced in [MSOI+02]. The importance of network motifs as information-processing modules was modeled theoretically in [SOMMA02] and [MA03], and verified experimentally in [KMP+01], [MZA03], [RRSA02], and [ZMR+04].

2.5.1 Definition

In the traditional sense, a **network motif** (or simply "motif") is a recurring, significant pattern of interaction. More recently, techniques have bren developed which identify patterns of interactions conserved across evolution [FNS+06, KSK+03], which may lead to a new method of identifying biologically significant network motifs. This thesis is concerned with motifs in the former sense, though some of the techniques developed in Chapter 3 are applicable to motif-finding in the latter sense.

A graph H is a **motif** of a network G if H appears as a subgraph of G significantly more frequently than in randomized versions of G. Similarly, a graph H is an **antimotif** if it appears significantly less frequently than in randomized versions of the original network. Whether a subgraph is a motif or antimotif is a more robust property of a network than the exact number of times it appears in the network. It is thus useful to define the **motif profile** of a network as the set of connected graphs up to a given size which are motifs, the set which are antimotifs, and the set which do not deviate significantly from the background model.

The frequency of a subgraph can be compared against several background models by altering the method of randomization (see §2.5.2). Perhaps the simplest relevant background model thought to be significant is to preserve the degree of each node. A very popular model, developed in [MSOI$^+$02], additionally preserves the distribution of $(k-1)$-node subgraphs when looking for k-node motifs. (Preserving the distribution of $(k-1)$-subgraphs turns out to be infeasible for $k > 4$. See §3.5.) The dK-series [MKFV06] (reviewed in §2.4, above) can also be used as a slightly more stringent version of Milo *et al.*'s background model. It is unclear at this time whether this model actually adds significant information into the background, or whether it is mostly redundant for the purposes of network motifs.

Network motifs are often thought to be the building blocks of networks. A small number of motifs – the feed-forward loop and the single-input module (Figure 2-2) – have been shown to perform significant information-processing roles both theoretically [MA03, SOMMA02] and experimentally [KMP$^+$01, MZA03, RRSA02, ZMR$^+$04]. In addition to these examples, several other biologically significant patterns of interactions are likely to have been missed by the current definition of network motifs, or patterns deemed significant by the current motif definition may not in fact biologically significant.

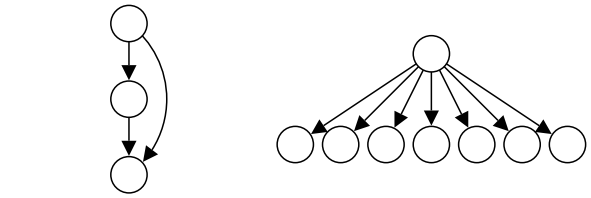


Figure 2-2: Two motifs whose dynamics have been studied both theoretically and experimentally: the feed-forward loop and the single-input module.

Recent work suggests that motifs are not necessarily functional, but may be simply by-products of evolution [SV06]. Additionally, there is experimental evidence [MIK$^+$04, MSOI$^+$02] that the networks of a similar nature have similar motifs. Thus determining the motifs of a network may give a clue as to the process by which it grows or evolves.

There is a hidden ambiguity in these definitions: it is important to specify what we mean by *subgraph*. In this work we consider only vertex-based subgraphs, which is the more straightforward condition because of the relationships between edge-based subgraphs. For example, consider the distribution of 3-node subgraphs shown in Table 2.3. Because each triangle contributes one triangle and three lines when counting edge-based subgraphs, it is possible that the line might be considered more or less significant when edge-based subgraphs are counted than when vertex-based subgraphs are counted. It is even possible for the line to be a vertex-based motif, but an edge-based *anti*motif (if the triangle were an antimotif).

		
Actual number vertex-based	1	19
Actual number edge-based	1	$19 + 3 \times 1$
Vertex-based in random ensemble	5 ± 1	7 ± 2
Edge-based in random ensemble	5 ± 1	$(7 + 3 \times 5) \pm \sqrt{2^2 + 3 \times 1^2}$
Vertex-based z-score	-4	6
Edge-based z-score	-4	0

Table 2.3: Edge-based subgraphs are assigned different levels of significance than vertex-based subgraphs because of the linear relationships between edge-based subgraphs. We use only vertex-based subgraphs.

2.5.2 Randomizing Networks

In the science of networks, a randomized version of a network – or a random network with similar properties to the original – is often needed as a null model. The three properties that are easiest to reproduce in a random graph are (in order): average degree, degree distribution, the degree of each node. In this work we use rewiring exclusively, which preserves the degrees of individual nodes.

An in-depth discussion of methods of randomization – including a rationale for using rewiring – can be found in [MKFV06].

Random rewiring of a network proceeds as follows. A pair of edges e_1, e_2 is chosen uniformly at random. If the edges do not share a common vertex, and if the comple-

mentary edges (see Figure 2.5.2) are not already present in the graph, then e_1 and e_2 are removed and the complementary edges are added. This preserves the degrees of all nodes involved. As with many Markov processes, it is unknown how quickly this process converges, but [GMZ03] shows that this process is an irreducible, symmetric, and aperiodic Markov chain which converges experimentally in $O(|E|)$ steps.

Figure 2-3: Degree-preserving rewiring. Random pairs of non-incident edges are chosen (as shown). If the solid edges are present and the dotted edges are not, these four nodes may be rewired by removing the solid edges and adding the dotted ones.

After a randomized version of the initial network has been obtained by rewiring, simulated annealing with rewiring can be used to simultaneously reproduce other properties of the original graph, such as the distribution of k-node subgraphs [MKFV06, MSOI$^+$02]. Unfortunately, preserving the distribution of k-node subgraphs is practically infeasible for $k > 3$. In §3.5 we propose two new background models to help alleviate this problem without relaxing the conditions on network motifs too much.

2.5.3 Graph Isomorphism

In Chapter 3 we introduce a method for counting all the instances of a graph as a subgraph of a network, with many applications to network motifs. In performing this counting, however, algorithms tend to find the same instance of a subgraph more than once, because of its symmetries. Graph isomorphism captures this notion of symmetry, and the notion of when two graphs are really "the same."

Two graphs G and H are said to be **isomorphic** if they have the same edges. In other words, if there is a map $f(v)$ from $V(H)$ to $V(G)$ such that $(v, w) \in E(H)$ if and only if $(f(v), f(w)) \in E(G)$. If such a map exists, it is called an **isomorphism** from H to G. For example, the two graphs in Figure 2-4 are isomorphic. An isomorphism

from the left graph to the right graph is given by associating each vertex on the left with the vertex of the same number on the right.

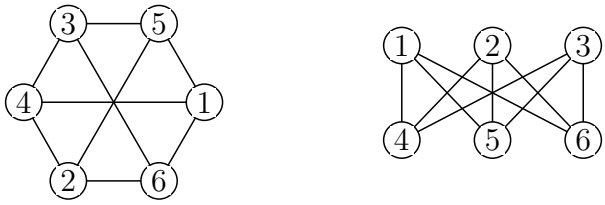


Figure 2-4: Two isomorphic graphs. The vertex labels denote an isomorphism.

As stated above, isomorphism is a form of equivalence. In particular, isomorphism is a transitive relation between graphs because the composition of two isomorphisms is again an isomorphism. Additionally, isomorphisms are invertible: for any isomorphism $f : G \to H$, there is an inverse isomorphism $f^{-1} : H \to G$ such that $f \circ f^{-1}$ is the identity map.

In general testing whether two graphs are isomorphic is computationally difficult. It is known that graph isomorphism is in the complexity class NP, though it is unknown whether graph isomorphism is either in P or NP-complete (in fact, graph isomorphism is one of the few problems thought to be in NP but *not* NP-complete, if $P \neq NP$). We will explore algorithms for isomorphism testing in §3.2.2.

An isomorphism from a graph to itself is called an **automorphism**. Note that every graph has the identity map as a trivial automorphism. Because automorphisms are also composable and invertible, the set of automorphisms of a graph has the mathematical structure of a *group* [Art91, HEO05, Ser03], and we sometimes refer to the **automorphism group** of a graph. See Appendix A for a brief review of the theory of graph automorphism groups. Automorphisms play an important role in the methods in Chapter 3, particularly in §3.3.2.

2.5.4 Limitations of Current Motif-Finding Methods

Current motif-finding methods are limited to motifs of very few nodes. This is due to a number of factors, discussed in more detail in Chapter 3, but perhaps the greatest limting factor is the size of the search space. The number of non-isomorphic connected graphs on n grows faster than exponentially in n, and the number of instances of such graphs in the undirected PPI net of *S. cerevisiae* [HBH$^+$04] appears to grow only slightly less than exponentially (see Table 2.4).

Nodes	Undirected	Directed	Instances in the undirected FYI network [HBH$^+$04]
3	2	13	11,881
4	6	199	69,865
5	21	9,364	408,295
6	112	1,530,843	2,280,781
7	853	8.8×10^8	12,353,532
8	11,117	1.8×10^{12}	66,493,797
9	261,080	1.3×10^{16}	—
10	1.1×10^7	3.4×10^{20}	—
11	1.0×10^9	3.2×10^{25}	—
12	1.6×10^{11}	1.1×10^{31}	—

Table 2.4: The number of non-isomorphic graphs represents the size of the search space for network motifs, and partially explains why finding larger motifs is so difficult. Here we also present the number of instances of subgraphs up to 8 nodes in the undirected protein-protein interaction network of *S. cerevisiae* [HBH$^+$04], determined using the new methods developed in Chapter 3
.

Exact counting methods have only been reported to find motifs up to 4 nodes [MSOI$^+$02] and motif generalizations up to 6 nodes [KIMA04b]. Subgraph sampling methods have found motifs up to 8 nodes [BP06, KIMA04a, MZW05]. Using the approach described in the next chapter, however, we are able to find a motif of 15 nodes, and explore subgraphs of 20 nodes (and potentially even larger subgraphs).

Chapter 3

A New Approach for Discovering Network Motifs

Network motifs – or frequently recurring circuits – have long been used in engineering computer chips, and have recently been shown to exist in more naturally occurring networks as well, such as food webs, the internet, and the protein interaction networks of various species [MSOI+02]. By analogy with their engineered counterparts, it is hoped that network motifs will allow us to understand these natural networks in terms of their fundamental computational building blocks.

Several studies have demonstrated both theoretically [SOMMA02, MA03] and experimentally [KMP+01, MZA03, RRSA02, ZMR+04] that network motifs can play crucial information processing roles in cellular networks. However, *in silico* models of network growth based on genetic mutation and duplication have also produced similar network motifs [KBL06], raising the question of whether network motifs are in fact functional, or are simply the by-products of the other evolutionary forces shaping the networks in question [SV06]. Even if they are "merely" by-products, studying network motifs can still provide insight into the processes by which networks grow and evolve.

A network motif is formally defined as a subnetwork that appears more frequently than by chance. Many different background models have been used to evaluate network motifs (see §3.5). Rather than hoping to find a background model which reveals

biologically meaningful motifs, we treat motifs as a language for describing the properties of a network. (Biologically relevant motifs are probably more likely to be found by including more biological information than simply the network structure, e.g. cellular dynamics and evoluationary history.) From this point of view, the background model and the motif distribution are complementary sources of information. The background model captures some information about the network (e.g. its degree distribution), and the network motifs capture the rest. The choice of background model is thus a trade-off between how hard it is to create an ensemble of networks under that model, and how much information about the network the model captures. In §3.5, we explore various background models, and introduce a new model which is both easily computable and captures more information than previous models.

Additionally, motif-finding methods can be applied to study networks more generally, based on their subgraphs. A particular type of subgraph introduced in [MKFV06] generalizes many important network properties that have been studied to date, namely the degree distribution, clustering coefficient, and degree correlations. We call this type of subgraph a **degree-correlated subgraph**, in which the degrees of the nodes in each instance of the subgraph are also taken into account. For example, one subgraph might be a triangle in which one node has degree 10 and the other two nodes have degree 5. Note that single-node degree-correlated subgraphs capture the degree distribution, two-node degree-correlated subgraphs capture degree correlations, and three-node degree-correlated subgraphs capture the clustering coefficients. It is important and useful to have a systematic set of properties for evaluating models of network growth – viz. model A reproduces the actual network up to size 2 degree-correlated subgraphs, but model B reproduces it up to size 3. Since degree-correlated subgraphs provide a systematic set of properties that encompass many informative network properties, having efficient algorithms to find these subgraphs is important for studying networks.

There are two basic methodologies for finding network motifs: exact counting (e.g. [MSOI$^+$02]) and subgraph sampling (e.g. [BP06, KIMA04a, MZW05]). Because exact counting is so computationally expensive [KIMA04a], subgraph sampling has proven

more effective at discovering larger motifs (up to 8 nodes, compared to 4 node motifs and 6-node motif generalizations with exact counting).

There are at least two important questions for the application of network motifs:

1. How can biologically meaningful circuits be discovered *in silico*, and perhaps used to guide experiment?

2. What are the larger structures that provide information and insight into a network's properties, and how can they be identified?

We propose several possible directions towards answering these questions, all of which seem to require a common tool: an algorithm to find all instances of a given subgraph (motif) in a network. Additionally, finding all instances of a motif allows us to explore motif clustering; in doing so, we may learn how dependent a motif's significance is on the accuracy of the experiments that produced the network.

Thus the aim of this chapter is to improve exact counting techniques so that they can be used to find all instances of a given subgraph, up to larger sizes than were achievable with previous methods.

3.1 Advantages of the New Approach

3.1.1 Querying Whether a Given Subgraph is a Motif

Because this new approach only searches for instances of a particular motif, rather than all motifs of a given size, it can be used to query whether a given subgraph is a significant motif. Because the algorithm need not find all subgraphs of a given size, query subgraphs can be much larger than motifs discovered using previous approaches. In particular, any subgraphs determined experimentally or suspected for other reasons to be significant can be queried, even if they are much larger than other network motifs reported to date.

3.1.2 Discovering Larger Motifs (Up to 15 Nodes) in Combination with Subgraph Sampling

By picking a random connected subgraph on n nodes and then finding all instances of that subgraph in the original network and in a random ensemble, the new approach can be used to find much larger motifs than before. In §3.4, we present the first network motif of 15 nodes, and we explore subgraphs of 10 and 20 nodes.

3.1.3 Applications to Motif Clustering (Up to 20 Nodes) and Network Alignment

Because the new approach finds all instances of a subgraph, rather than simply determining the significance of a motif (as with subgraph sampling), it can be used for further motif studies as well. In particular, the way in which instances overlap and interact – motif clustering – is easily explored. Based on the new, large subgraphs we explore in §3.4, subgraph clustering seems to be even *more* important for larger subgraphs than for smaller ones. Additionally, the new approach can be combined with network alignment either by using the discovered motif instances as seeds for alignment, or by finding larger motifs within aligned portions of networks.

3.1.4 Time and Space

Table 3.1 compares the running time of counting all subgraphs of a given size with our method to the time of counting all subgraphs of a given size with previous exact counting methods [MSOI$^+$02]. In our implementation of the previous exact counting method, we include all of the improvements listed in §3.2.2. Even with the improvements to the previous method, our new approach is still about 3.5 times faster.

The new method can also take considerably less space than current methods. Unlike current methods (§3.2.1), the new method does not need to keep track of which subgraphs it has encountered – this is taken care of automatically. Thus if the new method is used solely to count the number of instances, it saves considerable

space compared to current methods. If used to output a list of those instances, however, the new method must use the same amount of space as current methods to keep such a list.

Size	Time of new method	Time of method of [MSOI+02]	Speedup	Number of connected graphs	Instances of connected graphs
3	0.8	1.4	1.75x	2	11,811
4	3.0	11.3	3.75x	6	69,865
5	31	114	3.68x	21	408,295
6	462 (∼8 min)	1,541 (∼25 min)	3.33x	112	2,280,781
7	8,569 (∼143 min)	[Out of memory]	N/A	853	12,343,532

Table 3.1: Comparison of the running times of the original method of counting subgraphs [MSOI+02] and our new method, on the FYI dataset [HBH+04]. Unless otherwise stated, all times are in seconds. For seven nodes, the method of [MSOI+02] ran out of memory.

3.2 Comparison with Existing Methods of Exact Counting

The current method of exact counting in widest use is due to Milo, *et al.* [MSOI+02]. It is essentially a depth-first search through the space of connected subgraphs (not to be confused with a depth-first search on the network itself) to find all subgraphs of a given size n.

The algorithm loops through all vertices v. It treats v as a subgraph of one node. For each subgraph it has found so far, it loops through all possible extensions of that subgraph by one neighboring node. This process is repeated recursively until all n-node subgraphs are encountered. The number of subgraphs of each isomorphism type is then tallied.

Once the subgraphs are enumerated, counting them by isomorphism type can require extensive isomorphism testing. The simplest way to determine the number of subgraphs of each isomorphism type is to test each subgraph to see if it is isomorphic to any of the subgraphs encountered by the algorithm so far. Unfortunately, this

algorithm runs in time $O(NC)$ where N is the number of non-isomorphic graphs of size n – which grows faster than exponentially – and C is the number of instances of subgraphs of size n – which also grows quite rapidly (see Table 2.4).

In the §3.2.1, we show how to avoid the overcounting that results from walking a subgraph in multiple different ways. In §3.2.2, we present two improvements to this algorithm, which will also be useful in the new algorithm presented in §3.3.

3.2.1 Correcting for Overcounting

Note that a single subgraph can be discovered multiple times by the above algorithm, because of the multiple ways of walking the subgraph. For example, consider the triangle in Figure 3-1. This triangle will be discovered six times, when walked in the following orders: 1-2-3, 1-3-2, 2-1-3, 2-3-1, 3-1-2, 3-2-1. It may be tempting to see this list and think of the six symmetries of the triangle, but this list in fact arises for a different reason: the triangle is counted six times because it can be walked in six different ways, and not because it has six symmetries.

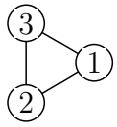


Figure 3-1: A triangle. The numbers represent the order in which the counting algorithm loops through vertices.

To correct for overcounting, the algorithm must keep track of exactly which subgraphs have been encountered so far. This is most efficiently done with a hash set, so that discovering duplicates only takes $O(1)$ time. Unfortunately, maintaining such a set takes $O(C)$ space, where C is the number of instances of subgraphs encountered (see the last column of Table 2.4). Although the algorithm now returns the correct counts, it still takes time counting each subgraph more than once.

To avoid some of the *time* spent overcounting, the algorithm should take advantage of the fact that it searches through the vertices in (an arbitrary) order. After all subgraphs of size n including vertex i have been discovered, vertex i should never

again be used.

But the triangle above will still get counted twice: 1-3-2, 1-2-3. It is tempting to say that only in-order walks should be considered, i.e. *not* 1-3-2 because 2 comes after 3 in the walk. Unfortunately, no such method can avoid overcounting without missing some subgraphs, e.g. the subgraph in Figure 3-2, and thus the visited list appears to be necessary.

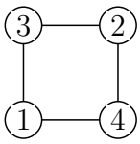


Figure 3-2: An example of why ordering the vertices cannot be used to avoid overcounting subgraphs. If the algorithm only counted subgraphs that were walked with their labels in order, this subgraph would never be counted.

3.2.2 Improving the Existing Methods

Any algorithm which counts the number of subgraphs exactly must take time which is at least proportional to the number of subgraphs. However, the algorithm of [MSOI+02], above, takes additional time doing many isomorphism tests. In this section, we aim to improve their algorithm by reducing the number of isomorphism tests performed and by improving the isomorphism tests themselves.

Although isomorphism is thought to be a difficult computational problem, it is important to have a relatively efficient isomorphism algorithm when counting subgraphs with the algorithm of [MSOI+02]. The simplest isomorphism algorithm tries all $n!$ possible maps between two graphs on n nodes and then checks to see if any of these maps preserves the structure of the graph.

Most improvements on this basic algorithm make use of **graph or vertex invariants**. An invariant is a property that is necessarily the same for isomorphic graphs. For example, the number of vertices, the number of edges, and the degree distribution are all graph invariants that are easily checked. If two graphs differ in any of these invariants, then they cannot be isomorphic.

All efficiently computable (i.e. in polynomial time) graph invariants known to date suffer from the fact that there are non-isomorphic graphs which have the same invariant. If there were an efficiently computable graph invariant that were additionally garaunteed to be distinct for non-isomorphic graphs, it would immediately provide an efficient graph isomorphism test, which is not known to exist.

There is a tradeoff between the complexity of computing an invariant and its discriminative power. For example, there are many non-isomorphic graphs with the same number of edges, but fewer non-isomorphic graphs with the same degree sequence. (See Table 3.2.) Another way of looking at it is that the number of edges correctly discriminates undirected graphs up to 3 nodes (i.e. the first pair of non-isomorphic graphs with the same number of edges has 4 nodes), while the degree sequence correctly discriminates undirected graphs up to 4 nodes.

A pair of invariants which involve computing all-pairs shortest paths can correctly discriminate non-isomorphic *directed* graphs up to 7 nodes [BP06], and correctly discriminates most directed graphs up to 8 nodes. However, for *undirected* graphs, these invariants do not provide a significant advantage over the degree distribution (see Table 3.2), particularly given their cost.

Nodes	Number of distinct values			
	Exact	$\|E\|$	Degree sequence	ℓ_1, ℓ_2 from [BP06]
3	2	2	2	2
4	6	4	6	6
5	21	7	19	20
6	112	11	68	76
7	853	16	236	269
8	11,117	22	863	1021

Table 3.2: The number of classes of undirected graphs discriminated by several different graph invariants. Each entry lists the number of distinct values of the invariant specified.

Hashing Based on Graph Invariants

To avoid the super-exponential factor of N, the number of non-isomorphic graphs on n nodes, in the runtime of the above algorithm, the algorithm can employ hashing

based on graph invariants. Rather than checking to see if each graph is isomorphic to any of the graphs seen so far, the algorithm need only see if each graph is isomorphic to any of the graphs seen so far *with the same hash value of the invariant*. Using a hash of the degree sequence, the hashtable lookup can be computed in linear time. Although Table 3.2 gives some sense of the utility of each invariant mentioned, Table 3.3 more directly measures the potential improvement gained by hashing graph invariants. Note that while the invariants of [BP06] had some advantage over the degree distribution in Table 3.2, they have absolutely no advantage over the degree distribution in terms of hashing. Thus we employ hashing based on the degree distribution. Table 3.4 shows the speed-up gained by hashing graph invariants.

Nodes	Maximum with same invariant			
	Exact	$\|E\|$	Degree sequence	ℓ_1, ℓ_2 from [BP06]
3	2	1	1	1
4	6	2	1	1
5	21	5	2	2
6	112	22	5	4
7	853	138	20	20
8	11,117	1579	184	184

Table 3.3: The maximum number of undirected graphs with the same invariant, for several different graph invariants. This shows that hashing based on graph invariants can reduce, e.g. a factor of 11,117 to a factor of 184 in the counting algorithm of [MSOI+02].

Nodes	Original	Hashing by Deg. Seq.	Speed-up
3	3.3 s	3.3 s	1.0x
4	13 s	10.4 s	1.25x
5	151 s	81.7 s	1.85x
6	4,821 s	1,698 s	2.84x

Table 3.4: Performance improvement by hashing graphs based on their degree sequences.

Using Vertex Invariants

Rather than trying all $n!$ possible combinations and seeing which ones preserve the graph structure, isomorphism testing can be significantly improved by taking advan-

tage of vertex invariants. The algorithm only checks maps which preserve the degree of each vertex, and the sequence of each vertex's neighbors' degrees. For example, if a graph has **one** vertex of degree one, **two** vertices of degree two, and **one** vertex of degree three, then rather than trying $4! = 24$ possible maps, by incorporating the vertex degrees, the algorithm need only try $\mathbf{1}! \cdot \mathbf{2}! \cdot \mathbf{1}! = 2$ possible maps.

This simple improvement to the basic isomorphism algorithm can have huge benefits. Table 3.5 lists the average number of isomorphims tried by the original algorithm, by including vertex degrees, and by also including neighbor degree sequences.

Nodes	Original	Using Degrees	Using Nbr. Degrees	Total Speed-up
3	6	3.13 ± 1.80	3.13 ± 1.80	1.9x
4	24	6.36 ± 6.11	6.36 ± 6.11	3.8x
5	120	13.0 ± 18.1	11.3 ± 18.7	10.6x
6	720	30.2 ± 58.3	24.5 ± 59.3	29.4x
7	5040	73.2 ± 186	49.2 ± 187	102x
8	40320	198 ± 649	107 ± 637	376x

Table 3.5: The average number of maps tested using the standard $n!$ isomorphism test, the isomorphism test taking into account vertex degrees, and the isomorphism test taking into account vertex degrees and the neighbor degree sequence of each vertex. The distributions are weighted by the number of occurrences in the protein-protein interaction network [HBH+04].

Additionally, the algorithm takes advantage of two vertex invariants. The algorithm only attempts to map x to y if x and y have the same degree, and if they have the same neighbor degree sequence.

3.3 Finding All Instances of a Subgraph

To find all instances of a query graph H in a network N, the algorithm attempts to map a copy of H wherever possible in the network. This is essentially the same as the isomorphism algorithm above, except it ensures that the subgraphs being searched are connected. In the language of artificial intelligence and search, this is a backtracking (depth-first) search *with forward checking* through the space of connected subgraphs.

The algorithm attempts to map H into the network N, ensuring that the graph structure is preserved at each step of the way. For each vertex $v \in H$ and each vertex

$w \in N$, the algorithm attempts to find an map f which maps $f(v) = w$. Let D be the domain of f and R its range, i.e. the vertices which have been mapped so far and their images, respectively. At each step, the algorithm attempts to extend f to some neighbor x of D and some neighbor y of R. This step succeeds if y is appropriately connected to R, i.e. if y is neighbors with $f(N(x))$ and not neighbors with $f(D - N(x))$ where $N(x)$ denotes the neighborhood of x. See Figure 3-3. This is the "forward checking" part of the algorithm, which effectively is an early abort. Without this, the algorithm would examine all connected n-node subgraphs, and then check whether each was isomorphic to H.

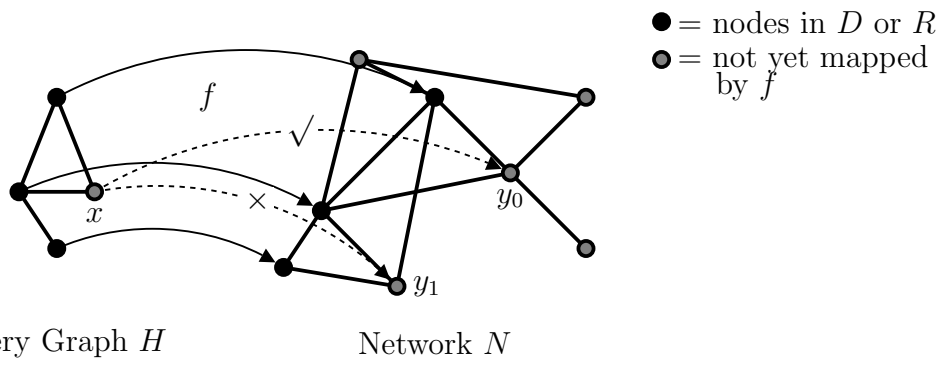


Figure 3-3: Vizualization of the subgraph-finding algorithm. At each step, the algorithm attempts to extend the partial map f to another vertex x, so that the resulting map continues to preserve the graph structure. Vertex y_0 is a possibility for $f(x)$, but vertex y_1 is not, because it is connected to the image of one of x's already-mapped *non*-neighbors.

Note that the forward checking is effectively an isomorphism test. However, because of the early aborts, the algorithm does not have to test every n-node subgraph against every possible isomorphism type. Forward checking was not possible with the previous exact counting method, because it was looking for *any* subgraph on n nodes, and not just one particular isomorphism type.

The algorithm also takes into account two vertex invariants: the vertex degree, and the ordered degree sequence of a vertex's neighbors. If a vertex $v \in H$ has degree d, the algorithm will not attempt to map it to a vertex w of degree less than d. A similar comparison is done on the neighbor degree sequence. Any ordered vertex invariant can also be used: that is, it must be that if the invariant of a vertex w in the

network is greater than the invariant of $v \in H$, then v could potentially be mapped to w.

3.3.1 Taking Advantage of the Degree Distribution

Most real-world networks have a scale-free, or at least a broad-tailed, degree distribution – i.e. they have many nodes with few neighbors and a few hubs with many neighbors. In such networks, hubs contribute the most to the combinatorial factors of the search. Because of the small path lengths and diameter (the FYI network [HBH+04] has diameter 25, and average path length 9.39 ± 3.58), it is likely that no matter where a search starts it will hit a hub. However, searches that start at hubs have many more possibilities.

Since the algorithm ignores any nodes it has already fully searched (see §3.2.1), it can effectively *delete* those nodes from the network for the remainder of the search. This reduces the degrees of the remaining nodes to be searched. Thus by starting with the low-degree nodes, the algorithm can reduce the degree of the high-degree nodes before they can contribute even more to the combinatorial search.

To take advantage of the degree distribution, the algorithm starts by sorting the nodes first by degree and then by the degree sequence of their neighbors. We have found empirically that this second tie-breaker provides a moderate additional performance improvement.

Since the algorithm now removes nodes once it has searched them fully, it is effectively changing the degrees of the remaining nodes. It might be fruitful to re-sort the remaining nodes. Empirically, however, this re-sorting ends up taking more time than it saves. Even simply re-inserting the altered nodes into sorted position takes more time than it saves, so it seems unlikely that re-sorting will provide an advantage. This advice should be taken lightly, however, as our results may be biased by the particular networks we are studying. Additionally, other re-sorting schemes (e.g. only re-sorting every 100 nodes) might improve performance.

3.3.2 Taking Advantage of Graph Symmetries

Unfortunately, the algorithm above counts each subgraph once for each symmetry it has. Since the algorithm does not keep a set of all subgraphs visited so far (unlike previous algorithms, see §3.2), it can find multiple maps from the query graph to the target network whose images are all the exact same subgraph. This factor can be corrected for by keeping a set of visited subgraphs (very space-intensive), or by dividing by the number of symmetries of the query graph – which can be exhaustively enumerated even for most graphs up to 20 or so nodes (and perhaps even larger using the techniques of [McK81]). However, to avoid the *time spent* overcounting, the algorithm can incorporate symmetry-breaking conditions into the forward checking.

To see how much this overcounting can slow down the algorithm, Table 3.6 lists some statistics on the number of automorphisms by graph size. The table also presents statistics which are weighted by the number of occurences of each graph in the FYI network [HBH+04]. The average number of automorphisms begins to decrease after size 6, because there are asymptotically more graphs with only the identity automorphism than with any other automorphism group. However, when weighted by the number of instances in the scale-free FYI network, the average number of automorphisms follows an exponential trend.

Nodes	# Graphs	Avg. # Aut's	Wtd. Avg.	Max # Aut's
3	2	4 ± 2	3.13 ± 1.80	6
4	6	7.67 ± 7.61	5.77 ± 5.93	24
5	21	11.52 ± 24.84	10.85 ± 18.63	120
6	112	14.73 ± 68.69	22.16 ± 58.04	720
7	853	13.29 ± 174.90	46.29 ± 186.2	5040
8	11,117	9.05 ± 386.63	96.24 ± 627.8	40320

Table 3.6: Statistics on the number of automorphisms of connected graphs by size. The number of automorphisms is the number of times a subgraph search will count a single instance of a subgraph. The distributions in the fourth column are weighted by the actual number of subgraphs with a given number of automorphisms in the FYI network [HBH+04].

The algorithm imposes a set of conditions on the graph that effectively removes all of its symmetries. Recall that each vertex in the network being searched is considered

to have a label, corresponding to the order in which it is searched in the outermost loop of the algorithm (which, according to §3.3.1 above is in order by degree). The symmetry-breaking conditions thus take the form: "The label of vertex $f(v)$ must be less than the labels of any of the vertices $f(w_0), \ldots, f(w_k)$," where f is a map from the query graph to the network, and $v, w_0, \ldots, w_k \in H$. We abbreviate this condition as $v < w_0, \ldots, w_k$.

To determine the necessary symmetry-breaking conditions, first the algorithm must determine the symmetries of the query graph H. This is very quick – it can be done for *all* (not "each") 11,117 8-node subgraphs exhaustively in about 30 seconds on a standard laptop – and is easily parallelizable.

Next, the algorithm determines the vertex orbits of H. An **orbit** is a set of vertices which can be mapped to one another via some set of automorphisms (see Appendix A for a brief review of the underlying mathematics of symmetries, groups, and orbits). If the set of automorphisms is not specified, the full automorphism group of H is implied. The algorithm picks an orbit consisting of more than one vertex, and then picks a vertex in that orbit to have the minimum label. This effectively fixes the vertex, and thus removes any automorphisms that do not fix the vertex. The algorithm repeats this process with the orbits under the remaining automorphisms, until all the orbits are size 1 (at which point the only automorphism remaining is the identity). See Figure 3-4 for an example of the symmetry-breaking process.

The algorithm is actually a bit more complicated than this, because nodes mapped in the outermost loop are treated differently than nodes in the recursive calls. A node mapped in the outermost loop can be considered fixed by any symmetries the algorithm will overcount by. Thus for each node $v \in H$, a seperate set of symmetry-breaking conditions must be found, starting by assuming that v is fixed. However, because nodes in the same orbit of H are effectively the same, the algorithm need only find a separate set of symmetry-breaking conditions for one representative from each orbit. Additionally, for each vertex v in the network, the algorithm need only start by assuming that each orbit representative (rather than each vertex of H), in turn, is mapped to v.

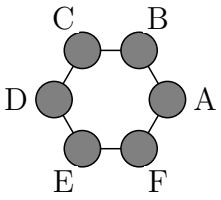
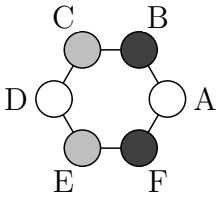
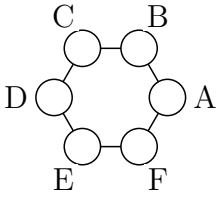
Graph	Orbits	Minimality Conditions
	{A,B,C,D,E,F}	none
	{A},{D},{B,F},{C,E}	$A < B, C, D, E, F$
	{A},{B},{C},{D},{E},{F}	$A < B, C, D, E, F$ $B < F$

Figure 3-4: Finding minimality conditions that will break all the symmetries of a 6-node graph. Nodes belonging to the same orbit under the automorphisms that obey the minimality conditions are shaded similarly, except for white nodes which are fixed.

3.4 Discovering Larger Motifs

To find larger motifs than is currently feasible by exact counting or subgraph sampling alone, we can combine the two using the algorithm presented above. Sample connected subgraphs of size n from either the real network or the random ensemble – using the random walk method of [MZW05] – and then use the algorithm of §3.3 to count how many there are. While this will not systematically discover all motifs of a given size, it is likely to discover some. Finding the query graph by sampling from the random ensemble makes it easier to find antimotifs, as otherwise they might appear so infrequently in the actual network that they would only get picked very rarely.

Additionally, any sub-networks discovered experimentally can be counted using the algorithm presented here.

We have employed the method suggested above, and have found one motif of 15

nodes (Figure 3-5). We have also used this method to explore the clustering properties of the 15-node motif, a subgraph of 10 nodes (Figure 3-6) and a subgraph of 20 nodes (Figure 3-7). We also found several graphs on 15 and 20 nodes that were relatively easy to count in the FYI network [HBH+04] but, because of their abundance, took very long to count in the random ensembles (too long to find all instances in 100 different randomized networks). We suspect that these are, therefore, antimotifs.

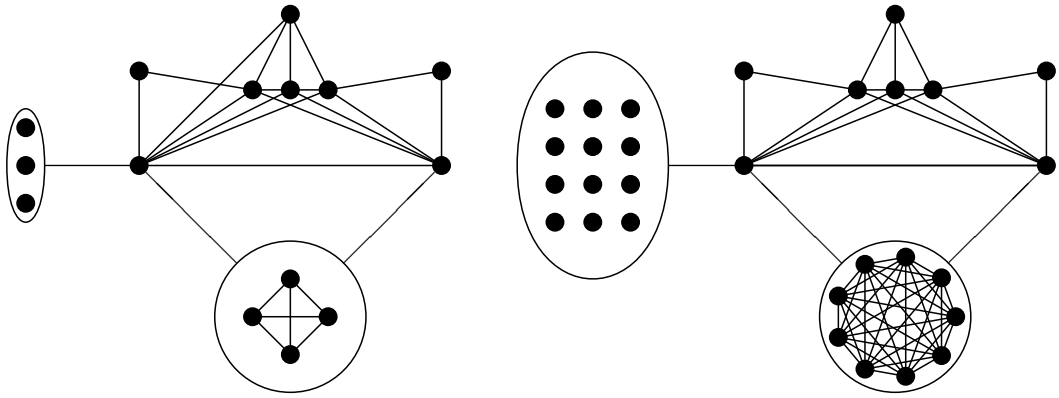


Figure 3-5: A motif of 15 nodes and 34 edges (left). An edge from a group of nodes to a node v indicates that each node in the group is connected to v. This motif appears 27,720 times in the FYI network [HBH+04], and does not appear at all in the random ensembles based on the degree distribution and the 3-node subgraph distribution. All 27,720 instances are clustered into a total of 29 nodes (right), yielding a subgraph clustering score (§3.4.1) of 19,454. Note that 3 nodes are chosen from the group on the left and four from the complete graph at bottom, yielding $\binom{12}{3}\binom{9}{4} = 27,720$ distinct instances.

3.4.1 Examining and Quantifying Motif Clustering

It has been observed that even the smallest subgraphs cluster together [VDS+04]. Furthermore, based on our experience with larger subgraphs, it would appear that combinatorial factors introduced by motif clustering play a much larger role with larger subgraphs than with smaller subgraphs. For example, all instances of the 15-node motif above cluster into a total of 29 nodes (Figure 3-5). Every instance of the 15-node motif shares the same core of 8 nodes, and the remaining nodes result from

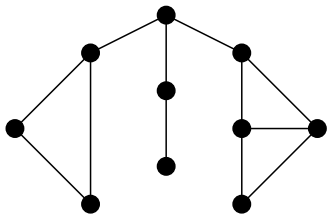


Figure 3-6: An anti-motif of 10 nodes and 12 edges. This appears 95,754 times in the FYI network [HBH+04], covering 472 proteins, yielding a subgraph clustering score (§3.4.1) of 18,858. It appears $2,596,601 \pm 640,778$ ($z = -3.9$) times on average in a random ensemble of 100 graphs with the same degree distribution as the FYI network, and $5,804,173 \pm 4,768,333$ ($z = -1.19$) times on average in a random ensemble of 100 graphs with the same degree distribution and distribution of 3-node subgraphs as the FYI network.

a choice of 3 out of 12 nodes (on the left in the figure) and 4 out of 9 nodes (the cluster at the bottom of the figure). The 20-node graph we examined has similar properties (Figure 3-7), but there are dependencies between which nodes are chosen for each instance of the graph because of connections between the attaching nodes (on the right in the figure).

In addition to this anecdotal evidence, we find that in the random ensembles these highly clustered subgraphs (the 15-node and 20-node subgraphs) either appear a combinatorial number of times, or not at all. For example, in the random ensemble of 100 graphs which preserve the degree distribution and the distribution of 3-node subgraphs, the 20-node subgraph appears 0 times in 92 out of the 100 graphs, but appears an average of $9,585 \pm 60,463$ times across the whole ensemble. (The 15-node motif does not appear *at all* in any of the random ensembles we examined.)

Quantifying the clustering of motifs may turn out to be an important aspect in identifying relevant motifs with more than ~ 8 nodes. We define the **subgraph clustering score** of a subgraph G in a network N is the average over all instances m of G in N of

$$\frac{1}{|V(G)|} \sum_{Vertices\ v \in m} |\{\text{Instances } m' | m' \neq m \text{ and } v \in m'\}|$$

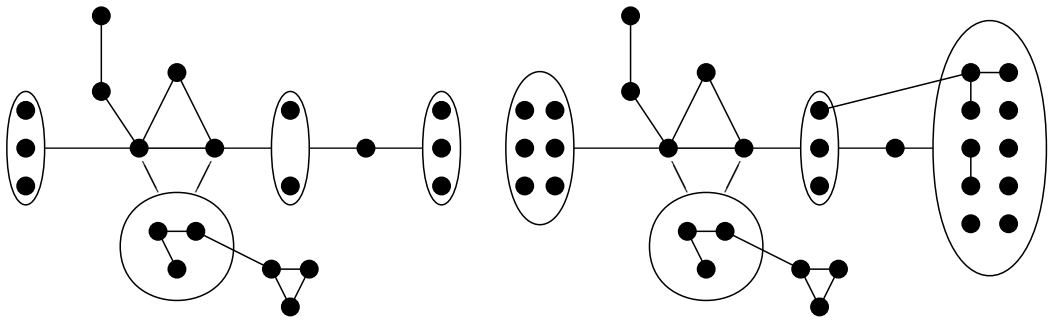


Figure 3-7: A subgraph of 20 nodes and 27 edges. An edge from a group of nodes to a node v indicates that each node in the group is connected to v. This motif appears 5,020 times in the FYI network [HBH+04], and $9,585 \pm 60,463$ ($z = -0.08$) times on average in a random ensemble of 100 graphs with the same degree distribution and distribution of 3-node subgraphs as the FYI network. All 5,020 instances are clustered into a total of 31 nodes, shown here, yielding a subgraph clustering score of 3,965.

The subgraph clustering score has the following nice properties:

- Scores of subgraphs of varying sizes can be meaningfully compared because the value is divided by the number of vertices in the subgraph;
- Subgraphs are not considered more clustered simply because they appear more frequently, since the index is the average over all instances (rather than, e.g. the sum);
- The more instances overlapping at any vertex, the higher the score;
- The more vertices shared by any two instances, the higher the score.

The graphs in Table 3.7 are ordered from intuitively "most clustered" to intuitively "least clustered," and the subgraph clustering score correctly puts the graphs in the desired order. Note that both the sum and the average correctly order the five toy examples (all of which have the same number of vertices and the same number of

instances), but only the average correctly orders the real examples (which have both different numbers of vertices and different numbers of instances).

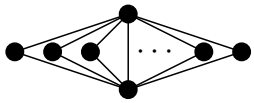
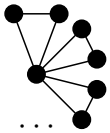
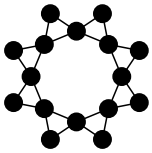
Motif and Graph	$\frac{1}{\lvert m\rvert}\sum_{v\in m}\lvert\{m' \neq m \lvert v \in m'\}\rvert$	
	Sum over motif instances m	Average over motif instances m
Motif: Figure 3-5, 15 nodes 27,720 instances in 29 nodes	0.54×10^9	19,454
Motif: Figure 3-6, 10 nodes 95,754 instances in 472 nodes	1.80×10^9	18,858
Motif: Triangle  n instances in $n+2$ nodes	253.33	12.66 $(=\frac{2}{3}(n-1))$
Motif: Triangle  n instances in $2n+1$ nodes	126.66	6.33 $(=\frac{1}{3}(n-1))$
Motif: Triangle  n instances in $2n$ nodes	13.33	0.66 $(=2/3)$
Motif: Triangle  n instances in $2.5n$ nodes	6.66	0.33 $(=1/3)$
Motif: Triangle  n instances in $3n$ nodes	0	0 $(=0)$

Table 3.7: The motif clustering score of two subgraphs of 10 and 15 nodes in the PPI network of *S. cerevisiae*and of four toy subgraphs. The order of the motif clustering scores for the toy subgraphs shows that the score agrees well with our intuition of clustering. All values here are reported for $n = 20$ for the toy subgraphs, with the score as a function of n in parentheses.

3.4.2 Biological Relevance

The 29-node subgraph which encompasses all instances of the 15-node motif (Figure 3-5) comprise a biologically meaningful part of the PPI net of *S. cerevisiae*. The proteins corresponding to the 8 node "core" shared by all instances of the 15-node motif, and the 11-node complete graph (including the two hubs in the core) are listed in Table 3.8. The 12 attachments are: YOR250C (CLP1), YPR115W, YER013W (PRP2), YPL089C (RLM1), YFL033C (RIM15), YLR228C (ECM22), YNL216W (RAP1, GRF1, TBA1, TUF1), YDR259C (YAP6, HAL7), YIL129C (TAO3, PAG1), YKL012W (PRP40), YDR207C (UME6, CAR80, NIM2, RIM16), and YGR097W (ASK10).

8-node Core		11-node Complete Graph	
Systematic Name	Common Names	Systematic Name	Common Names
YDR167W*	TAF10, TAF23, TAF25	YDR448W	ADA2, SWI8
YGL112C*	TAF6, TAF60	YDR176W	NGG1, ADA3, SWI7
YML015C	TAF11, TAF40	YBR081C	SPT7, GIT2
YCR042C	TAF2, TAF150, TSM1	YLR055C	SPT8
YGR274C	TAF1, TAF130, TAF145	YOL148C	SPT20, ADA5
YML09W	TAF13, FUN81, TAF19	YGR252W	GCN5, ADA4, SWI9
YPL129W	TAF14, SWP29, TAF30, TFG3, ANC1	YDR392W	SPT3
YMR227C	TAF7, TAF67	YPL254W	HFI1, ADA1, SUP110, SRM12, GAN1
		YHR099W	TRA1

Table 3.8: The proteins involved in all instances of the 15-node motif of Figure 3-5. The 8-node core is the transcription factor TFIID complex; the 11-node complete graph (including the two proteins marked with a * from the core) is the SAGA complex; and the 12 attachments – activators and suppressors – are listed in the text.

The 11-node complete graph is the SAGA complex, and almost all of its proteins are involved in chromatin modification and histone acetylation; the 8-node core is the

transcription factor TFIID complex; and the 12 attachments are known activators and suppressors of these two complexes [LY00]. Using our new approach to motif-finding, we have re-discovered the cellular transcription machinery based solely on the structure of the protein interaction network.

3.5 Background Models

The background model most in use today is a random ensemble of graphs with the same degree distribution as the network being studied. Random graphs with this property are easily computed by edge swapping (see §2.5.2). In fact, all of the background models discussed here will preserve this property, as they are based on edge swapping followed by simulated annealing with edge swapping (so at every step the degrees of each vertex are preserved).

Milo *et al.* [MSOI$^+$02] additionally preserve the distribution of $(k-1)$-node subgraphs when identifying motifs of size k. Unfortunately, due to correlations induced by edge swapping and local minima that are not surmountable using simulated annealing, this is practically infeasible for $k > 4$.

Rather than preserving the distribution of $(k-1)$-node subgraphs, we propose a background model in which only the distribution of 3-node subgraphs is preserved (along with the degree distribution, as always), regardless of the size k of the motifs being identified.

This simple (and easily computable) background model seems to capture much of the information in the motif profile against a degree-preserving background model. Using the new algorithm of §3.3, we counted all connected subgraphs with between 3 and 7 nodes in the protein-protein interaction network of *S. cerevisiae* using the FYI dataset [HBH$^+$04], and two random ensembles of 100 networks each (one preserving only the degree distribution, the other also preserving the distribution of 3-node subgraphs). Note that there are 994 isomorphism types of connected graphs of these sizes.

Using a significance cutoff of $|z| \geq 2.0$, 657 of the 994 subgraphs were significant against the degree-preserving background, while only 111 were significant against the 3-node-subgraph-preserving background. Using a significance cutoff of $|z| \geq 10.0$, these numbers are 216 and 32, respectively. (It is interesting to note that all of the subgraphs significant at the $|z| > 10.0$ level against the 3-node-subgraph-preserving distribution are motifs, and not antimotifs.) Data and further details can be found

in Appendix B.

One way to incorporate more information into the background model is to use the dK-series ([MKFV06] and §2.4) up to $d = 3$ node degree-correlated subgraphs.

Since it is believed that whether or not a subgraph is a (anti)motif is more important than the exact number of occurences, we propose another background model: preserve the motif profile of the graph. We define the motif profile as the set of subgraphs which are motifs, the set which are antimotifs, and the set which are not significant against some other background model. This definition is recursive, as the 5-node motif profile relies on the 4-node motif profile, etc.

We also generated a random ensemble of 100 graphs with the same 4-node motif profile as the FYI network [HBH+04], where the significance of the 4-node subgraphs was evaluated against the exact distribution of 3-node subgraphs. We did not include the distribution of 3-node subgraphs in the background for this ensemble, though it is feasible to do so. Without the distribution of 3-node subgraphs, preserving the 4-node motif profile yielded similar results to preserving *only* the distribution of 3-node subgraphs as above, at the $|z| \geq 10$ level. There were 31 significant subgraphs against the 4-node motif profile and 32 against the 3-node-subgraph distribution, though which were significant and which were over- or under-represented differs between the two background distributions.

3.6 Discussion and Future Directions

In this chapter we have presented an algorithm for finding all instances of a given subgraph. This algorithm can be used to find all subgraphs of a given size faster and with less memory than the method of [MSOI+02], which is the standard method of exact counting to date. Additionally, by counting only one subgraph at a time, the new method is easily paralellizable – an important property as cluster computing is becoming cheaper and more popular in the biological and other sciences.

Recall the two questions posed in the introduction:

1. How can biologically meaningful circuits be discovered *in silico*, and perhaps

used to guide experiment?

2. What are the larger structures that provide information and insight into a network's properties, and how can they be identified?

In §3.6.3, we propose a method to answer Question 1 by combining network alignment and network motifs. Question 2 can be answered either by looking for larger motifs as in §3.4 and §3.6.2, or by developing a language to describe larger, more flexible network structures, such as the motif generalizations of [KIMA04b], and searching for those. Before these proposals, however, §3.6.1 suggests ways this algorithm could be applied to even larger structures.

3.6.1 Further Applications of the New Algorithm

One of the limiting factors of the current algorithm is the amount of space required to store all the symmetries of the query graph, though for most graphs on 20 nodes, and possibly larger, the current algorithm suffices. To search for even larger structures, however, this bottleneck would have to be improved.

To remove this bottleneck, the algorithm could find only the *generators* of the automorphism group of the query graph H, and note the whole group. (For a brief review of group theory and more on generators, see Appendix A.) The generators of the automorphism group are a set of automorphisms S such that any automorphism can be written as a composition of automorphisms from S. Such a generating set can be found using the methods in [McK81]. The algorithm then needs the sequence of subgroups attained by fixing various vertices. These can be obtained by assigning an appropriate order to the vertices, and then finding a strong generating set [HEO05, Ser03] for the automorphism group. A strong generating set can be found in $O(|V(H)|^2)$ time, and has the property that it is easy to find generators of the subgroup which fixes a particular set of vertices.

3.6.2 Finding Even Larger Motifs

As with sequence motifs, it may be possible to take already-known network motifs of a given size and extend them to larger motifs. Simply pick an already-known motif, extend it by one or more vertices, and then use the new approach presented in this chapter to find all instances of it in the network being studied and in the random ensemble being used for motif-finding. Using either subgraph sampling or exact counting this was impossible, as determining the significance of the extended subgraph is just as hard as finding larger motifs to begin with, in these previous approaches.

3.6.3 Combining Network Alignment and Network Motifs

By analogy with sequence motifs, one attempt to answer Question 1 would involve finding subgraphs that are not only more frequent than expected, but also conserved across multiple species. The approach presented in §3.4 could be used to find large, meaningful structures in the aligned portion of two or more networks. See Figure 3-8.

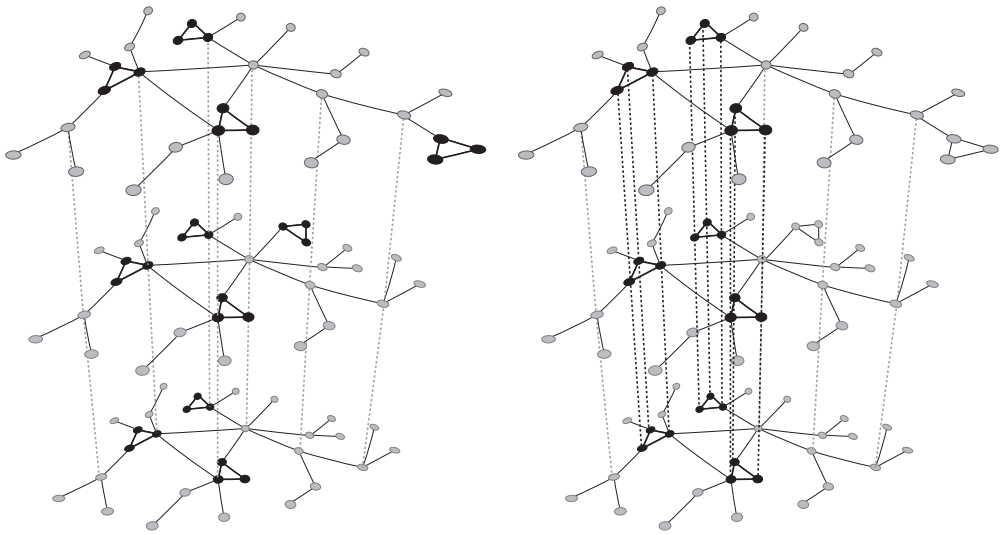


Figure 3-8: One way of combining network motifs and network alignment is to only count *conserved instances* of subgraphs when determining their significance as motifs (as is standard in searching for sequence motifs). After aligning several networks (left), search for significant subgraphs in the aligned portions of the networks (right). Note that the triangles on the right in the uppermost graph and in the back of the middle graph are not counted, as they is not conserved across species.

Additionally, one could find all instances of large network motifs using the approach of §3.4, and then use these instances to help seed the network alignment. (Currently network alignment seeding is typically based solely on cross-species information, such as homology, and the network structure is not taken into account until after the seeding phase of the alignment.)

Chapter 4

Biologically Grounded Models of Network Growth

(This chapter is related to joint work with Alexei Vázquez, Manolis Kellis, Matt Rasmussen, and Albert-László Barabási, to be submitted.)

4.1 Models of Network Growth

Models of network growth are often designed by incorporating one or more features known to be relevant to the network being studied, and then incorporating other features or optimizing parameters in an attempt to get the statistics (§2.3) of the model to agree with the actual network.

In biological networks, this has resulted in models of network growth that incorporate preferential attachment [Bar03], genetic duplication and divergence [CLDG03, IKMY05, KBL06, PEKK06, PSSS03], and sometimes even whole genome duplications, e.g. [WG05].

But in the age of whole genomes, model parameters for biological network growth can be determined by *observation*, rather than by fitting. Previous modelling techniques assume the model is correct, and fit the parameters to the data. But if a model with observed parameter values reproduces the properties of the actual network, then we know we are on the right track.

Additionally, for some networks the actual history of growth is available. For example, in the movie actors' network [IMD06] – in which nodes are actors that are connected by an edge if they have performed in a movie together – the date of each movie puts a timestamp on when each link was created. For such networks, models can be developed by actually *watching the network grow!*

We propose a model of protein interaction network growth based on binding domains. For the purposes of the model, a domain is considered atomic, and all interactions must be mediated by a pair of interacting domains. The domain-domain interactions are modelled as an Erdős-Rényi random graph, though any model can be substituted for this, and the resulting protein interaction network studied. Each protein is modelled as a linear chain of domains, where a single type of domain can appear more than once in the chain. The "library" of domains can be either finite or infinite – we allow it to be infinite. The three operations of the model are: gene/protein duplication (since the domains are copied, the protein interactions are automatically duplicated), domain loss, and domain creation. The model is validated by reproducing properties of the observed protein interaction network when using the observed parameter values. In the next section we show how these three parameters can be estimated using phylogenetic data.

4.2 Estimating the Parameters

We estimate the three main parameters (genetic duplication, domain loss, and domain creation) using data from seven closely related yeast species: *S. cerevisiae*, *S. paradoxus*, *S. mikatae*, *S. bayanus*, *K. waltii*, *K. lactis*, and *A. gossypii*. The parameters were each estimated in terms of number of events from the common ancestor of *A. gossypii* and *S. cerevisiae* (the root of the phylogenetic tree containing these seven species) to *S. cerevisiae*, and then normalized for use in the model. This part of the phylogenetic tree corresponds to roughly 0.11 substitutions per site.

4.2.1 Rate of Gene Duplication

We reconstructed 1083 gene trees and reconciled them to the seven-species yeast tree using the methods of [RK06]. We normalize the depth of each gene tree to the median depth of any of the present-day genes (often this is the most evolved gene from the four species closest to *S. cerevisiae*). The rate of gene duplication is the average number of duplications per substitution per site, over the period of time where there is enough meaningful data (i.e. until the normalized depth of 1 – see Figure 4-1). There are approximately 1715 gene duplication events over this time period.

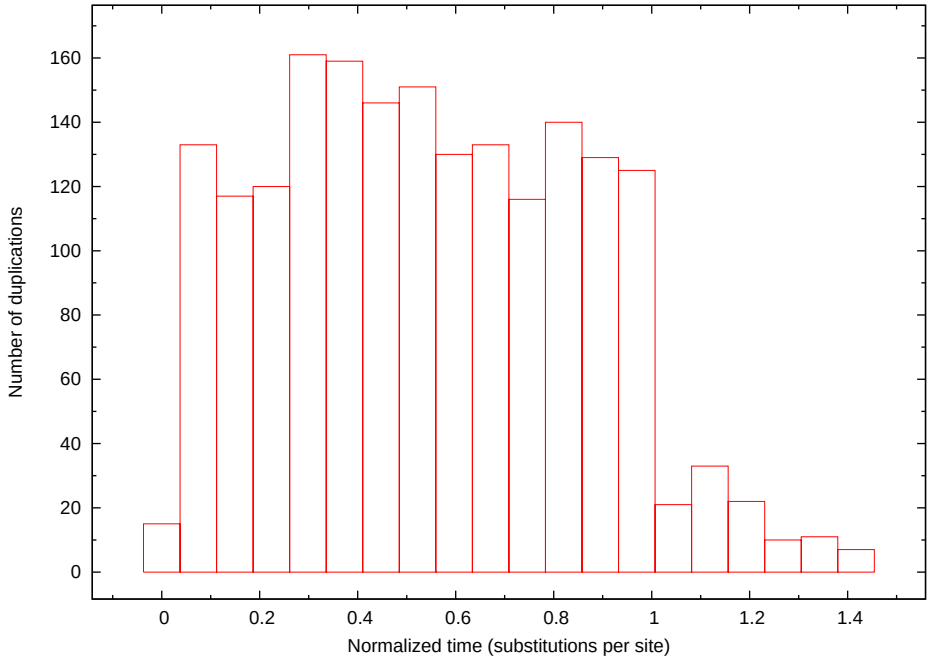


Figure 4-1: Number of gene duplications per substitutions per site. The sharp dropoff at 1 SPS is because the depth of the gene trees was normalized to 1, which roughly corresponds to the most evolved gene in the four species closest to *S. cerevisiae*.

4.2.2 Rate of Domain Loss

A protein domain is considered present in a protein if 40% of the domain sequence is present in the protein sequence without many gaps in the alignment [CL86]. Thus if more than 60% of a domain's amino acids are modified (i.e. more than 0.6 substitutions per site), the domain is considered lost. Since *S. cerevisiae* is 0.11 SPS from

the root of the seven-species tree, we estimate that one in every 6 domains is lost, or approximately 290 domains.

We have not yet taken into account any additional evoluationary pressure to conserve binding domains, and this might give a more accurate measure of the rate of domain loss.

4.2.3 Rate of Domain Creation

To determine the rate of domain creation, we aligned the PFAM [BCD+04] domains in *S. cerevisiae* to their orthologs in *A. gossypii* using CLUSTALW [THG94]. Orthology was determined using the method of [RK06]. As above, we considered a domain present in *A. gossypii*– and thus in the common ancestor of *A. gossypii* and *S. cerevisiae*– if more than 60% of the domain sequence was conserved in the alignment. The number of domain creation events is the number of domains present in *S. cerevisiae* but not in *A. gossypii*. We ensured this count was independent of homology: i.e. a domain that was present in several homologs in *S. cerevisiae* but not in their ortholog in *A. gossypii* was considered a single domain creation event (presumably followed by one or more single gene duplications). There were 229 and such events.

Although the rate of domain creation is less than the rate of domain loss, the model can still produce a network with many domains because protein duplication is the driving force.

Chapter 5

Asymmetric Divergence of Duplicates and the *K. waltii* Interactome

(This chapter represents joint work with Jean-François Rual, Manolis Kellis, Albert-László Barabási, and Marc Vidal, in preparation.)

Genetic duplication, whether at the scale of a single gene, a chromosomal segment, or a whole genome, is a significant mechanism in evolution [Ohn70]. Single gene duplicates have been identified in *S. cerevisiae* for many years, and more recently it was identified that *S. cerevisiae* underwent a whole genome duplication (WGD) about 30 million years ago [KBL04]. When duplication occurs, it is believed that the two members of a duplicate pair initially have identical functions and interactions. Afterwards, there is a transient period during which one of the duplicates will differentiate, diverge in function, or disappear altogether [Wag02]. Exactly how this divergence occurs is still an open question, though it has been studied both theoretically [BLW04, IKMY05, PSSS03, Wag03] and observationally [ZLKK05].

For the purposes of this chapter, we use protein interactions as an indication of function.

In the next section, we discuss some of the theoretical work that has been done to date on the process of divergence after duplication. Then in §5.2 we propose a set of

experiments that will help shed light on this process more directly. The experiments are currently being undertaken in the lab of Jean-François Rual and Marc Vidal at the Dana Farber Cancer Institute of Harvard, and will be completed after the submission of this thesis.

5.1 Duplication and Divergence

After duplication, the two members of the duplicate pair can each gain or lose interactions, including the auto-interaction between the two duplicates themselves. See Figure 5-1.

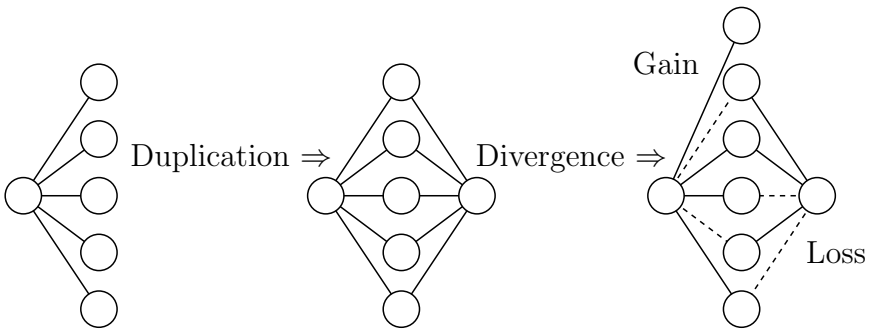


Figure 5-1: A single gene duplicates, along with all of its protein-protein interactions, and then the two duplicates differentiate and diverge.

Note that an interaction partner which only interacts with a single member of the duplicate pair could have been gained by one partner, or lost by the other. Divergence after duplication is said to be **symmetric** if the probabilities of gain and loss are both equal to 50%, and asymmetric if these probabilities deviate significantly from 50%.

There are two main theories as to how the functions of the duplicates evolve after duplication: neofunctionalization and subfunctionalization. A pair of duplicates is said to have undergone **neofunctionalization** if one of the pair maintains the function of the ancestral gene and the other takes on one or more new functions. A pair of duplicates is said to have undergone **subfunctionalization** if each member of the pair maintains a different part of the function of the ancestral gene. Typically

it is assumed that every function of the ancestral gene is taken up by at least one member of the pair.

Note that whether an instance of duplication and divergence is symmetric or not is independent of whether it is an instance of neofunctionalization or an instance of subfunctionalization.

By comparing the results of stochastic computational experiments in which the ancestral network is modelled probabilistically with observed network data, Wagner [Wag02] showed that asymmetric divergence occurs much more frequently than symmetric divergence. Additionally, [Wag02] provides a theoretical analysis suggesting that functions (interactions) are least likely to be lost from the species entirely in the maximally asymmetric case.

Since a WGD provides a wealth of examples of duplicated genes, all of which duplicated at the same time, it makes sense to study duplication and divergence by examining the ohnologs and their interaction partners. An **ohnolog** – named for Susumu Ohno, who first proposed genetic duplication as a significant process in evolution [Ohn70] – is a gene that was duplicated in a WGD event.

Amongst the ohnologs in the protein interaction network of *S. cerevisiae*, we find a correlation between the rate of sequence divergence and the degree of the protein in the PPI net. (We use the rates of sequence divergence calculated in [KBL04].) The ohnologs can be divided into three classes based on the rate of sequence divergence:

1. Ohnolog pairs that have diverged at relatively similar rates from their common ancestor ("undifferentiated"),

2. Ohnologs that have diverged significantly faster than their duplicate partners, and

3. Ohnologs that have diverged significantly slower than their duplicate partners.

In general, the second two classes come paired: that is, one ohnolog partner is rapidly diverging and the other is slowly diverging. However, in the FYI network [HBH+04], not all proteins from *S. cerevisiae* are present, and so the number of rapidly diverging ohnologs differs from the number of slowly diverging ohnologs.

Ohnolog pairs that have diverged at different rates tend to have lower degree than ohnolog pairs that have diverged at similar rates. Furthermore, amongst the ohnolog pairs diverging at different rates, the rapidly diverging ohnologs have lower degree than the slowly diverging ohnologs. See Table 5.1.

Divergence Class	Average Degree	Number present in FYI network [HBH$^+$04]
Rapid	1.9	19
Slow	2.6	40
Undifferentiated	4.7	126
All ohnologs	4.0	185
All proteins	3.6	1379

Table 5.1: The correlation between sequence divergence and number of protein interactions in the ohnologs of *S. cerevisiae*. In addition to the three classes of ohnologs based on sequence divergence, the table also includes statistics for all ohnologs together, and for all proteins in the FYI network [HBH$^+$04]. For each set of proteins studied here, the degree distribution roughly follows a power law.

5.2 Proposed Experiment

We propose to experimentally explore protein-protein interactions in *K. waltii*, a pre-WGD ancestor of *S. cerevisiae*. For each interaction in *S. cerevisiae* involving at least one ohnolog, there are four possible interactions to test:

1. *S. cerevisiae* ohnolog with *S. cerevisiae* interaction partner
2. *S. cerevisiae* ohnolog with *K. waltii* ortholog of *S. cerevisiae* interaction partner
3. *K. waltii* ortholog of *S. cerevisiae* ohnolog with *S. cerevisiae* interaction partner
4. *K. waltii* ortholog of *S. cerevisiae* ohnolog with *K. waltii* ortholog of *S. cerevisiae* interaction partner

Experiment 1 duplicates previous work, but would ensure that all the data is coming from consistent experimental techniques. Experiment 2 allows us to determine if an interaction was gained or lost due to changes in the interaction partner since the WGD. Experiment 3 allows us to determine if an interaction was gained or lost due to

changes in the duplicated gene. Experiment 4 allows us to determine if an interaction was gained or lost since the WGD, and also provides the first glimpse into the protein-protein interaction network of a pre-WGD yeast species. For reasons of cost, we limit our experiments to the most informative regarding asymmetric divergence of duplicates: experiment 3.

Based on the union of two high-confidence datasets [GAG+06, HBH+04] we selected 683 interactions to test, involving 407 *S. cerevisiae* proteins and 129 *K. waltii* proteins (see Appendix C for the complete list). Orthology was determined based on [KBL04].

The FYI dataset [HBH+04] consists of 1379 proteins and 2493 interactions (its largest connected component consists of 778 proteins and 1798 interactions). We modified the Gavin, et al. dataset [GAG+06] by taking only those interactions whose "socio-affinity score" – a log-odds ratio developed in [GAG+06] – was above 6.5, resulting in a network with 1204 proteins and 3512 interactions (its largest connected component consists of 781 proteins and 2968 interactions). We chose the threshold of 6.5 by comparing the socio-affinity scores of edges in the FYI dataset with the socio-affinity scores of edges that are not present in the FYI dataset (see Figure 5-2). Although $\sim$4.5 would provide the maximum likelihood cutoff, we chose the higher values of 6.5 based on the highest score of edges not present in the FYI dataset, since the FYI dataset is significantly based on yeast two-hybrid experiments, which are known to have more false positives than false negatives.

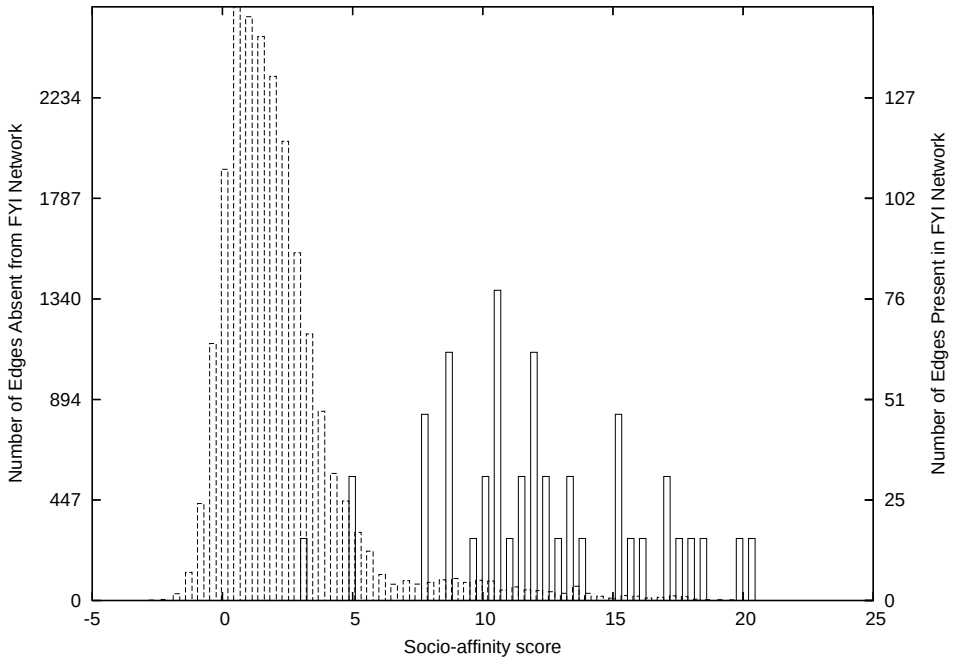


Figure 5-2: Frequency of socio-affinity scores [GAG$^+$06] of the 1272 edges present in the FYI network [HBH$^+$04] and of the 22,341 edges absent from the FYI network. The scale for each group (edges present and edges absent) is normalized to a percentage of the group total for the purposes of visual comparison.

Chapter 6

Contributions

In this thesis we have explored the structure of real-world networks and the evolution of protein interaction networks, using the yeast *S. cerevisiae* as a model organism.

Chapter 5 sets out to explore experimentally the process of genetic duplication and divergence by examining the interactions involving pairs of ohnologs, and their pre-WGD ancestors. By seeing if the pre-WGD ancestral proteins from *K. waltii* interact with the post-WGD proteins in *S. cerevisiae*, we will be able to distinguish between cases of neofunctionalization and subfunctionalization in the ohnologs of *S. cerevisiae*. The yeast two-hybrid experiments are currently underway, and will be completed after the submission of this thesis. These are the first experiments to probe the relationship between the protein interactions in two species separated by a WGD.

In Chapter 4, we introduce a new model of protein interaction network growth, based on the underlying binding domains of the proteins. Our main contribution here is to estimate parameters for the model based on phylogenetic evidence, rather than fitting the model parameters to observed network data. Unlike previous modelling techniques, this ensures that we do not consider our model validated solely because it manages to overfit the observed data.

In Chapter 3, we introduce a new approach to finding network motifs. This new approach is significant for any network science, not just biological networks. Rather than determining the significance of all motifs of a given size, as the previous

methods of subgraph sampling and exact enumerating do, we present an algorithm to count all instances of a particular subgraph (§3.3). By using this algorithm on all non-isomorphic subgraphs of a given size, the new approach can recapitulate the previous exact counting approach. Despite the costs of counting all instances of each subgraph one at a time, rather than altogether, our algorithm manages to outperform previous exact counting methods by a factor of ~ 3.5 (even when we improve the exact counting algorithm by the standard but important techniques in §3.2.2) by taking advantage of the scale-free degree distribution (3.3.1) and the symmetries of the query graphs (3.3.2). Neither of these adaptations can be used in previous exact counting techniques because of the structure of the algorithms.

In §3.5, we introduce two new background models against which network motifs can be identified. We show that one of these models – preserving the distribution of 3-node subgraphs, regardless of the size k of the motifs being sought – is effectively computable *and* captures much more information about the network than the degree distribution alone. Against this background model, there are very few motifs smaller than six nodes, suggesting that the degree distribution and the distribution of triangles in the PPI net of *S. cerevisiae* contain nearly all the relevant information regarding subgraphs up to 5 nodes.

The ability to find all instances of a given subgraph means that our new approach has many more applications than previous approaches. The new approach can be combined with subgraph sampling to find larger (anti)motifs than ever before, by picking a random large connected subgraph and then finding all instances of it in the network being studied and in a random ensemble of related networks. In §3.4, we use this technique and present the first ever 15-node motif, and we explore 10- and 20-node subgraphs. Additionally, the 15-node motif is biologically relevant: it is part of the cellular transcription machinery.

The new approach can also be used to study motif clustering. Based on the larger motifs found in §3.4, we believe that studying motif clustering may be even more important for large subgraphs than for the smaller subgraphs found by previous methods. In §3.4.1 we introduce the subgraph clustering score, which allows the

clustering of subgraphs of differing sizes to be meaningfully compared. We show that the subgraph clustering score makes sense both for a carefully constructed series of graphs that can be clearly ordered in terms of how clustered they are intuitively, and for comparing the 10-node subgraph explored in §3.4 to the new 15-node motif.

Finally, in §3.6, we discuss further possible application of the new approach:

- finding larger significant structures which are regular expression-like generalizations of subgraphs, rather than fixed subgraphs;
- finding still larger motifs by extending current motifs and then counting all instances of the resulting extension;
- combining network alignment and network motifs by searching for large structural motifs in the aligned portion of multiple networks; and
- combining network alignment and network motifs by using all instances of significant motifs to help seed the alignment process.

Appendix A

Graph Symmetries: A Group Theory Primer

A.1 Introduction and Definitions

In §3.3.2, we present part of an algorithm for counting subgraphs which effectively "removes" all the symmetries of a graph. At first glance, it might appear to many computer scientists (and others) that this might require $\log_2(n)$ symmetry-breaking conditions if the graph has n symmetries. But in fact the relationship between the symmetries is more complicated than that, and often all of the symmetries of a graph can be broken with many fewer conditions. This appendix briefly reviews some basic *group theory*, which is the mathematical theory of symmetries. Determining exactly how many symmetry-breaking conditions are necessary serves as a guiding example.

As mentioned in §2.5.3, the automorphisms of a graph G form a group, denoted $Aut(G)$. An automorphism of a graph G is a map $f : G \to G$ such that if (v, w) is an edge of G then $(f(v), f(w))$ is also an edge of G, and vice versa. The automorphisms of a graph have the following properties:

1. The composition of two automorphisms is again an automorphism.

2. The composition of automorphisms (as with the composition of any functions) is associative: $(f \circ g) \circ h = f \circ (g \circ h)$.

3. Every graph has the identity map id as a (trivial) automorphism. Furthermore, composing any automorphism with the identity leaves it unchanged: $id \circ f = f \circ id = f$.

4. Every automorphism f has a unique inverse f^{-1}, since automorphisms are one-to-one and onto functions. In particular, $f \circ f^{-1} = f^{-1} \circ f = id$.

These four properties – closure, associativity, identity, and inverses – are the defining properties of a group.

Abstractly, a group is any set with a binary operation (in the case of graph automorphisms the binary operation is functional composition) which satisfies the above properties. Very often, regardless of the binary operation, multiplicative notation is used. So rather than writing $f \circ g$ for the composition of two automorphisms, write fg. Note that this means "first apply g, then apply f," as in the composition of functions. Additionally, write f^n for the n-fold composition of f with itself. The usual rules of integer exponentiation apply ($f^a f^b = f^{a+b}$, f^0 is defined to be the identity, etc.) Finally, the group identity is often written e or 1.

To clearly specify the underlying set of a group and the binary operation, write e.g. $(Aut(G), \circ)$. This notation is often abused however, and the name of the underlying set is also typically used as the name of the group.

Because groups contain inverses, cancellation is allowed. In other words, if $ab = ac$ in a group G, then multiplying by a^{-1} on the left yields $b = c$. Using cancellation, it is not hard to show that the identity is unique, and that each element of a group has only one inverse.

In finite groups, $f^{-1} = f^n$ for some n. Consider the automorphisms of a finite graph G. If $f \in Aut(G)$, then so is f^n for any n. But because $|Aut(G)|$ is finite, $f^i = f^j$ for some i, j (otherwise the sequence $f, f^2, f^3, \ldots$ would contain infinitely many distinct elements of $Aut(G)$). Then, since f is invertible we can compose both of these maps with f^{-i-1}, to get $f^{-1} = f^{j-i-1}$.

A **cyclic group** consists of all the powers of a single element, and is defined entirely by its size. The abstract cyclic group of n elements is denoted C_n.

Other examples of groups you are probably familiar with are:

- $(\mathbb{Z}, +)$, the integers under addition,
- $(\mathbb{Z}/n\mathbb{Z}, +)$, the integers modulo n under addition,
- $(\mathbb{C} - \{0\}, \times)$, the non-zero complex numbers under multiplication (zero has no multiplicative inverse),
- $(\mathbb{Z}/p\mathbb{Z} - \{0\}, \times)$, the non-zero integers modulo a prime p under multiplication, and
- $GL_n(\mathbb{C})$, the set of all invertible $n \times n$ complex matrices under matrix multiplication.

Note that matrix multiplication and composition of functions are *non-commutative*, i.e. it is not necessarily the case that $ab = ba$.

A **subgroup** is simply a subset of a group which is itself a group with respect to the same operation as its parent. Write $H < G$ to denote that H is a subgroup of G (the notation is purposefully distinct from subset notation). This is a very important concept for studying groups, and in particular for our aim of finding symmetry-breaking conditions on graphs. Each symmetry-breaking condition imposed effectively reduces the automorphism group to one of its subgroups.

To determine how many symmetry-breaking conditions are necessary to remove all symmetries of a graph, it is necessary to understand something about the relationship between the size of a group and the size of its subgroups. The size of a group $|G|$ is called its **order**. The first theorem taught in almost all, if not all, courses on group theory is **Lagrange's theorem**: The order of a subgroup H divides the order of its parent group G.

To see why, consider the **left cosets** of H, which are all the sets $aH = \{ah | h \in H\}$ for $a \in G$.

Step 1. Because of cancellation, each coset is the same size as H.

Step 2. Every element of G lies in some coset of H. Since $1 \in H$, $a \in aH$ for every $a \in G$.

Step 3. Any two cosets aH, bH are either identical or disjoint. Suppose two cosets aH and bH are not disjoint. Then there is some element $c \in aH \cap bH$. Let $c = ah_1 = bh_2$, for some $h_1, h_2 \in H$. Then $b^{-1}a = h_2 h_1^{-1}$. Since H is a subgroup, it is closed under multiplication and inversion, and thus $b^{-1}a = h_2 h_1^{-1} \in H$, and also $(b^{-1}a)^{-1} = a^{-1}b \in H$. But then any element $bh \in bH$ is equal to

$$\begin{aligned} bh &= b(b^{-1}aa^{-1}b)h \\ &= (bb^{-1})a(a^{-1}bh) \\ &= a(h_1 h_2^{-1} h) \in aH \end{aligned}$$

so $bH \subset aH$. By symmetry , $bH = aH$. Thus the left cosets of H partition G.

Step 4. Since the left cosets of H all have size $|H|$, and they partition G, $|H|$ evenly divides $|G|$.

The **index** of a subgroup H in a group G is the number of distinct left cosets of H, and is denoted $[G : H]$. By the proof above, $|G| = [G : H]|H|$, which is the more traditional form of the statement of Lagrange's theorem.

For any set of automorphisms S, the group **generated by** S, denoted $\langle S \rangle$ is the smallest group containing S. If the resulting group is finite, it is easy to see that it simply consists of all possible compositions of the automorphisms in S.

By Lagrange's theorem, any group of prime order has no subgroups other than itself and the trivial subgroup $\{1\}$. In fact, it turns out that every group of prime order is cyclic. Suppose G is a group of prime order, and let a be any non-identity element of G. If $G \neq \langle a \rangle$, then $\langle a \rangle$ is a nontrivial cyclic subgroup of G, contradicting the fact that $|G|$ is prime.

A.2 Orbits and Symmetry-Breaking

The automorphisms of a graph G are said to *act on* the vertices of G. Each automorphism is essentially just a permutation of the vertices. For a set of automorphisms S and a vertex v, the **orbit** of v under S is the set of vertices to which v can be

mapped by the automorphisms in S. Note that the orbit of v under S and the orbit of v under $\langle S \rangle$ are identical if $\langle S \rangle$ is finite.

The **stabilizer** of a vertex v in a group of automorphisms G is the set of all automorphisms which fix v, denoted $Stab_G(v)$ or G_v. G_v is a subgroup of G since (a) the identity fixes v, (b) the composition of two automorphisms fixing v again fixes v, and (c) the inverse of an automorphism fixing v also fixes v. In particular, $|G_v|$ divides $|G|$.

Each symmetry-breaking condition used in §3.3.2 effectively reduces the automorphism group from H to H_v. Let $H = Aut(G)$ for some graph G. Thus the search for symmetry breaking conditions corresponds to finding a chain of groups $H > H_{v_0} > H_{(v_0,v_1)} > \cdots > \{1\}$. Since the order of each stabilizer divides the order of its parent, each step in this chain reduces the size of the remaining group by at least 2, validating the intuition that it takes at most $\log_2(|Aut(G)|)$ steps to break all the symmetries. More generally, the number of symmetry-breaking conditions can be estimated by examining the prime factorization of the number of automorphisms of a graph.

In order to need $\log_2(|Aut(G)|)$ symmetry-breaking conditions, each stabilizer must have index 2 in the previous stabilizer, and in parituclar $|H| = |Aut(G)|$ must be a power of 2. Asymptotically, there are many groups of order 2^n (typically there are more groups of order 2^n than there are groups of any other order with exactly n prime factors). However, there are not nearly so many graph automorphism groups of order 2^n, particularly for small graphs.

A.2.1 Minimizing the Number of Symmetry Breaking Conditions

Group theory can also help optimize the number of symmetry breaking conditions by providing a means to choose which orbit to break up next. The **orbit-stabilizer theorem** says that $|H| = |Orbit(v)||Stab(v)|$, or equivalently $[H : H_v] = |Orbit_H(v)|$.

Step 1. $[H : H_v] \geq |Orbit_H(v)|$. For any $g \in H_v$ and any $a \in H$, $ag(v) = a(v)$.

So every element of the left coset aH_v sends v to the same element of its orbit. Thus there are at least as many left cosets as there are elements in v's orbit.

Step 2. $|Orbit(v)| \geq [H : H_v]$. If $a(v) = b(v)$, then ab^{-1} fixes v, and so $ab^{-1} \in H_v$. Thus $aH_v = bH_v$, so there are no more left cosets of H_v than the number of elements in v's orbit.

Thus to maximize $[H : H_v]$ at each step in the chain corresponding to symmetry-breaking, the algorithm should choose to break up the largest orbit first.

Appendix B

All Motifs up to 7 Nodes With Exact Counting in the PPI Network of *S. cerevisiae*

Using the exact counting method of Chapter 3, we determined the significance of every subgraph up to 7 nodes in the PPI net of *S. cerevisiae*[HBH$^+$04] against two different ensembles: one which preserves only the degree distribution, and one which additionally preserves the distribution of 3-node subgraphs (i.e. triangles).

The list is sorted in terms of the significance (z-score) of the subgraph against the distribution of triangles. We have displayed only those subgraphs which have a z-score greater than 4.0 in magnitude. The complete list can be found at http://compbio.mit.edu/networks/.

It is interesting to note that, of the most significant subgraphs, many more of them are motifs (51) than antimotifs (3) against the background distribution of triangles. At the $|z| > 10.0$ level, the only significant subgraphs are motifs.

Additionally, the background model which includes the distribution of triangles captures much more of the graph structure than the background model which only preserves the degrees. At the $|z| > 2.0$ level, 657 subgraphs are significant against the degree distribution, while only 111 subgraphs are significant against the distribution of triangles and degrees. (Note there are only 994 isomorphism types of connected

undirected graphs with between 3 and 7 nodes.) At the $|z| > 4.0$ level, these numbers become 392 and 54, respectively, and at the $|z| > 10.0$ level, 216 and 32.

Finally, we highlight the importance of specifying the background model and/or the significance cutoff when calling a subgraph a motif or an anti-motif. Table B.1 displays the number of graphs which are considered e.g. motifs against the degree distribution and anti-motifs against the degree and triangle distribution for $|z| > 2.0$. For $|z| > 4.0$, there are only 3 graphs which switch from anti-motif to motif and 3 vice versa, and for $|z| > 10.0$ there are is only 1 graph which switches from motif to anti-motif, and none in the other direction.

		Degree and Triangle Background		
		Anti-motif	Insignificant	Motif
Degree Background	Anti-motif	8	386	13
	Insignificant	13	313	10
	Motif	26	183	41

Table B.1: The number of graphs which have different status as motifs against the two background models considered, e.g. motif against one background model and anti-motif against the other.

	Original	Degrees	Degrees & Triangles
Graph	Count	Motif? (z-score)	Motif? (z-score)
	7,015	Motif (51.9)	Motif (1,607)
	1,237	Motif (103)	Motif (123)
	288,288	Motif (83,363)	Motif (75.4)
	15,057	Motif (174)	Motif (68.9)
	2,419	Motif (446)	Motif (53.8)
	2,147	None (2.76)	Motif (53.8)
	556	Motif (12.7)	Motif (44.6)

	64,555	Motif (26,481)	Motif (42.6)
	125,192	Motif (1,365)	Motif (41.2)
	300,743	Motif (5,210)	Motif (38.5)
	4,378	None (-2.49)	Motif (37.1)
	1,358	None (0.17)	Motif (31.7)
	20,120	Motif (54.4)	Motif (27.1)
	4,972	Motif (5.47)	Motif (24.9)
	22,930	Motif (31.1)	Motif (24.8)
	1,801	Motif (947)	Motif (24.4)
	18,175	Motif (648)	Motif (23.8)
	14,703	Motif (88.3)	Motif (22.9)
	252,604	Motif (89.3)	Motif (21.6)
	6,685	None (-3.40)	Motif (21.2)
	6,461	None (-0.21)	Motif (20.7)
	609,862	Anti (-13.2)	Motif (18.7)
	182,118	Motif (4.39)	Motif (15.3)
	14,833	None (0.92)	Motif (15.0)
	147,227	Motif (420,425)	Motif (13.4)

	4,994	None (-3.89)	Motif (13.4)
	2,676	None (-3.32)	Motif (12.6)
	142,221	None (2.83)	Motif (11.8)
	479	None (-2.36)	Motif (11.7)
	178,974	None (-3.81)	Motif (11.5)
	12,875	Motif (2,153)	Motif (11.0)
	4,702	None (-0.13)	Motif (10.1)
	196,270	Motif (68.5)	Motif (9.56)
	5,922	Motif (278)	Motif (8.21)
	94,313	Motif (9,714)	Motif (8.16)
	1,624	Motif (131)	Motif (7.92)
	155,299	Motif (372)	Motif (7.90)
	33,359	Motif (3,164)	Motif (7.63)
	278	Motif (4.13)	Motif (7.42)
	12,993	Anti (-9.45)	Motif (7.39)
	136,187	Motif (541)	Motif (6.84)
	81,582	Motif (21.2)	Motif (5.61)
	7,914	None (-3.75)	Motif (5.51)

	20,573	Motif (1,729)	Anti (-5.01)
	1,512	None (-0.23)	Motif (5.01)
	672	None (1.26)	Motif (4.93)
	22,730	Motif (100)	Motif (4.91)
	12,102	Motif (547)	Anti (-4.54)
	387	Motif (32.9)	Motif (4.51)
	1,513	None (-3.52)	Motif (4.41)
	4,109	Motif (11.8)	Anti (-4.39)
	209,776	Anti (-20.1)	Motif (4.32)
	92,372	Motif (82.9)	Motif (4.19)
	81,947	None (1.88)	Motif (4.15)

Appendix C

Protein-Protein Interactions Being Explored in *K. waltii* and *S. cerevisiae*

Protein-protein interactions being explored in *K. waltii* and *S. cerevisiae*. FYI indicates the FYI dataset [HBH+04] and TAP indicates the Gavin, et al. dataset [GAG+06]. The interactions to test are based on reported interactions between *S. cerevisiae* ohnologs (second column) and other *S. cerevisiae* proteins (third column). We test the *K. waltii* ancestor (first column) of the *S. cerevisiae* ohnologs against the *S. cerevisiae* interaction partners in order to determine if an interaction was gained or lost because of changes in the ohnologs after duplication. This data can also be found in text-only format (i.e. machine-parseable but still human-readable) at http://compbio.mit.edu/networks/, along with the ORF sequences used for each protein in the experiment, and is included here only for completeness.

K. wal. ancestor of ohnolog(s)	*S. cer.* ohnolog(s)	*S. cer.* interaction partner	Dataset
Kwal_92	YFR013W	YAR007C	TAP
Kwal_92	YFR013W	YBL003C	TAP
Kwal_92	YFR013W	YBR245C	TAP
Kwal_92	YFR013W	YFR037C	TAP
Kwal_92	YFR013W	YGL133W	TAP
Kwal_92	YFR013W	YKR001C	TAP

Kwal_92	YFR013W	YOL004W	TAP
Kwal_92	YFR013W	YPL082C	TAP
Kwal_96	YOL016C, YFR014C	YBR109C	FYI
Kwal_99	YLR258W	YIL045W	FYI
Kwal_99	YLR258W	YLR273C	FYI
Kwal_167	YLR249W	YBR118W	FYI
Kwal_167	YLR249W	YPR080W	FYI
Kwal_250	YCL011C	YDL014W	TAP
Kwal_250	YCL011C, YNL004W	YDR138W	TAP
Kwal_250	YCL011C, YNL004W	YHR167W	TAP
Kwal_250	YCL011C, YNL004W	YNL139C	TAP
Kwal_250	YNL004W	YML062C	TAP
Kwal_1015	YGR239C	YDR142C	FYI
Kwal_1019	YGR238C	YHR158C	FYI
Kwal_1019	YHR158C	YGR238C	FYI
Kwal_1382	YGR034W, YLR344W	YBL027W	FYI
Kwal_1382	YGR034W, YLR344W	YBL087C	FYI
Kwal_1382	YGR034W, YLR344W	YBL092W	FYI
Kwal_1382	YGR034W, YLR344W	YBR031W	FYI
Kwal_1382	YGR034W, YLR344W	YBR084C-A	FYI
Kwal_1382	YGR034W, YLR344W	YDL136W	FYI
Kwal_1382	YGR034W, YLR344W	YDL191W	FYI
Kwal_1382	YGR034W, YLR344W	YDR012W	FYI
Kwal_1382	YGR034W, YLR344W	YDR418W	FYI
Kwal_1382	YGR034W, YLR344W	YEL054C	FYI
Kwal_1382	YGR034W, YLR344W	YER117W	FYI
Kwal_1382	YGR034W, YLR344W	YGL103W	FYI
Kwal_1382	YGR034W, YLR344W	YGL135W	FYI
Kwal_1382	YGR034W, YLR344W	YGL147C	FYI
Kwal_1382	YGR034W, YLR344W	YGR085C	FYI
Kwal_1382	YGR034W, YLR344W	YIL018W	FYI
Kwal_1382	YGR034W, YLR344W	YIL133C	FYI
Kwal_1382	YGR034W, YLR344W	YJL177W	FYI
Kwal_1382	YGR034W, YLR344W	YKL180W	FYI
Kwal_1382	YGR034W, YLR344W	YLR075W	FYI
Kwal_1382	YGR034W, YLR344W	YLR340W	FYI
Kwal_1382	YGR034W, YLR344W	YNL067W	FYI
Kwal_1382	YGR034W, YLR344W	YNL069C	FYI
Kwal_1382	YGR034W, YLR344W	YOL127W	FYI
Kwal_1382	YGR034W, YLR344W	YOR063W	FYI
Kwal_1382	YGR034W, YLR344W	YPL131W	FYI
Kwal_1382	YGR034W, YLR344W	YPL220W	FYI
Kwal_1382	YGR034W, YLR344W	YPR102C	FYI
Kwal_1436	YDL175C	YJL050W	TAP
Kwal_1436	YDL175C	YLR347C	TAP

Kwal_1436	YDL175C	YMR125W	TAP
Kwal_1436	YDL175C	YNL251C	TAP
Kwal_1436	YDL175C	YPL190C	TAP
Kwal_1436	YIL079C, YDL175C	YOL115W	FYI,TAP
Kwal_1812	YBR216C	YML007W	FYI
Kwal_2150	YLL021W	YER149C	FYI
Kwal_2150	YLR313C	YBL016W	FYI
Kwal_2150	YLR313C, YLL021W	YDL159W	FYI
Kwal_2150	YLR313C, YLL021W	YLR319C	FYI
Kwal_2150	YLR313C, YLL021W	YLR362W	FYI
Kwal_2150	YLR313C, YLL021W	YOR231W	FYI
Kwal_2150	YLR313C, YLL021W	YPL140C	FYI
Kwal_2313	YLL016W	YLR310C	FYI
Kwal_2313	YLR310C	YAL005C	FYI
Kwal_2313	YLR310C	YBL075C	FYI
Kwal_2313	YLR310C	YDL047W	TAP
Kwal_2313	YLR310C	YER103W	FYI
Kwal_2313	YLR310C	YGL197W	TAP
Kwal_2313	YLR310C	YJL098W	TAP
Kwal_2313	YLR310C	YLL016W	FYI
Kwal_2313	YLR310C	YLL024C	FYI
Kwal_2313	YLR310C	YPL240C	FYI
Kwal_2313	YLR310C, YLL016W	YNL098C	FYI
Kwal_2421	YJL076W	YDL042C	FYI
Kwal_2421	YKR010C	YOL006C	FYI
Kwal_2935	YNL096C	YOR361C	TAP
Kwal_2935	YNL096C	YPR041W	TAP
Kwal_2935	YOR096W	YDR091C	TAP
Kwal_3650	YHL034C	YGR162W	TAP
Kwal_3650	YHL034C	YMR230W	TAP
Kwal_3650	YHL034C	YOL139C	TAP
Kwal_3747	YGR085C	YFR031C-A	TAP
Kwal_3747	YGR085C, YPR102C	YBL027W	FYI
Kwal_3747	YGR085C, YPR102C	YBL092W	FYI
Kwal_3747	YGR085C, YPR102C	YBR084C-A	FYI
Kwal_3747	YGR085C, YPR102C	YDL136W	FYI
Kwal_3747	YGR085C, YPR102C	YDL191W	FYI
Kwal_3747	YGR085C, YPR102C	YGL103W	FYI
Kwal_3747	YGR085C, YPR102C	YGR034W	FYI
Kwal_3747	YGR085C, YPR102C	YIL018W	FYI
Kwal_3747	YGR085C, YPR102C	YLR075W	FYI
Kwal_3747	YGR085C, YPR102C	YLR344W	FYI
Kwal_3747	YGR085C, YPR102C	YOL127W	FYI
Kwal_3747	YGR085C, YPR102C	YOR063W	FYI
Kwal_3747	YGR085C, YPR102C	YPL131W	FYI

Kwal_3747	YPR102C	YER006W	TAP
Kwal_3747	YPR102C	YPL093W	TAP
Kwal_3932	YBR118W	YKL081W	FYI
Kwal_3932	YBR118W	YPL048W	FYI
Kwal_3932	YPR080W, YBR118W	YAL003W	FYI
Kwal_3932	YPR080W, YBR118W	YLR249W	FYI
Kwal_3992	YGR092W	YAL021C	FYI
Kwal_3992	YGR092W	YDL160C	FYI
Kwal_3992	YGR092W	YNR052C	FYI
Kwal_3992	YGR092W, YPR111W	YIL106W	FYI
Kwal_4144	YDL075W	YDR060W	TAP
Kwal_4144	YDL075W	YDR101C	TAP
Kwal_4144	YDL075W	YNL110C	TAP
Kwal_4144	YDL075W, YLR406C	YJL189W	FYI
Kwal_4144	YDL075W, YLR406C	YMR242C	FYI
Kwal_4144	YDL075W, YLR406C	YOR312C	FYI
Kwal_4144	YLR406C	YHR052W	TAP
Kwal_4144	YLR406C	YNR053C	TAP
Kwal_4240	YDL061C, YLR388W	YGL123W	FYI
Kwal_4240	YDL061C, YLR388W	YHL015W	FYI
Kwal_4240	YDL061C, YLR388W	YHR203C	FYI
Kwal_4240	YDL061C, YLR388W	YJR123W	FYI
Kwal_4240	YDL061C, YLR388W	YJR145C	FYI
Kwal_4240	YDL061C, YLR388W	YNL178W	FYI
Kwal_4240	YDL061C, YLR388W	YOL040C	FYI
Kwal_4569	YDL226C	YDR264C	FYI
Kwal_4733	YIL133C, YNL069C	YDL136W	FYI
Kwal_4733	YIL133C, YNL069C	YDL191W	FYI
Kwal_4733	YIL133C, YNL069C	YDR395W	FYI
Kwal_4733	YIL133C, YNL069C	YGL103W	FYI
Kwal_4733	YIL133C, YNL069C	YGR034W	FYI
Kwal_4733	YIL133C, YNL069C	YIL018W	FYI
Kwal_4733	YIL133C, YNL069C	YLR075W	FYI
Kwal_4733	YIL133C, YNL069C	YLR344W	FYI
Kwal_4733	YIL133C, YNL069C	YNL301C	FYI
Kwal_4733	YIL133C, YNL069C	YOL120C	FYI
Kwal_4733	YIL133C, YNL069C	YOL127W	FYI
Kwal_4733	YIL133C, YNL069C	YOR063W	FYI
Kwal_4733	YIL133C, YNL069C	YPL131W	FYI
Kwal_4911	YIL113W	YHR030C	FYI
Kwal_4925	YIL109C	YDR517W	TAP
Kwal_4925	YIL109C	YLR208W	TAP
Kwal_4925	YIL109C, YNL049C	YPL085W	TAP
Kwal_4925	YIL109C, YNL049C	YPR181C	TAP
Kwal_5298	YOL115W	YDL175C	TAP

Kwal_5298	YOL115W	YFL008W	FYI
Kwal_5298	YOL115W	YFR031C	FYI
Kwal_5298	YOL115W	YIL079C	FYI
Kwal_5298	YOL115W	YNL251C	TAP
Kwal_5298	YOL115W	YPL190C	TAP
Kwal_5576	YHR030C	YIL113W	FYI
Kwal_5576	YHR030C	YOR231W	FYI
Kwal_5576	YHR030C	YPL140C	FYI
Kwal_5764	YER054C	YER133W	FYI
Kwal_5764	YER054C, YIL045W	YBR045C	FYI
Kwal_5764	YIL045W	YLR258W	FYI
Kwal_5799	YIL052C, YER056C-A	YDR395W	FYI
Kwal_5799	YIL052C, YER056C-A	YHL001W	FYI
Kwal_5799	YIL052C, YER056C-A	YKL006W	FYI
Kwal_5807	YER059W, YIL050W	YPL031C	FYI
Kwal_5887	YER070W	YIL066C	FYI
Kwal_5887	YER070W	YML058W	FYI
Kwal_5887	YIL066C	YER070W	FYI
Kwal_5887	YIL066C, YER070W	YGR180C	FYI
Kwal_5887	YIL066C, YER070W	YJL026W	FYI
Kwal_5953	YER081W	YIL074C	FYI
Kwal_5953	YIL074C	YER081W	FYI
Kwal_6006	YDL179W, YDL127W	YDR388W	FYI
Kwal_6006	YDL179W, YDL127W	YPL031C	FYI
Kwal_6069	YPL256C, YMR199W	YBR135W	FYI,TAP
Kwal_6069	YPL256C, YMR199W	YBR160W	FYI,TAP
Kwal_6069	YPL256C, YMR199W	YDL132W	FYI
Kwal_6225	YCR052W	YBR245C	TAP
Kwal_6225	YCR052W	YDR303C	TAP
Kwal_6225	YCR052W	YFR037C	FYI,TAP
Kwal_6225	YCR052W	YGR056W	TAP
Kwal_6225	YCR052W	YIL126W	TAP
Kwal_6225	YCR052W	YKR008W	TAP
Kwal_6225	YCR052W	YLR033W	TAP
Kwal_6225	YCR052W	YLR321C	FYI,TAP
Kwal_6225	YCR052W	YLR357W	TAP
Kwal_6225	YCR052W	YML127W	TAP
Kwal_6225	YCR052W	YMR072W	TAP
Kwal_6225	YCR052W	YMR091C	TAP
Kwal_6225	YCR052W	YPR034W	TAP
Kwal_6225	YCR052W, YNR023W	YMR033W	TAP
Kwal_6225	YNR023W	YBR289W	FYI,TAP
Kwal_6225	YNR023W	YHL025W	TAP
Kwal_6225	YNR023W	YJL176C	TAP
Kwal_6225	YNR023W	YML007W	TAP

Kwal_6225	YNR023W	YPL016W	TAP
Kwal_6325	YCR073C	YMR117C	FYI
Kwal_6325	YCR073C, YNR031C	YLR006C	FYI
Kwal_6344	YCR073W-A	YER133W	TAP
Kwal_6344	YCR073W-A	YLR028C	TAP
Kwal_6373	YGR118W, YPR132W	YGL123W	FYI
Kwal_6373	YGR118W, YPR132W	YHL015W	FYI
Kwal_6373	YGR118W, YPR132W	YJR123W	FYI
Kwal_6373	YGR118W, YPR132W	YNL178W	FYI
Kwal_6373	YGR118W, YPR132W	YOL040C	FYI
Kwal_6805	YAL051W	YOR363C	FYI
Kwal_6805	YOR363C	YAL051W	FYI
Kwal_6945	YAL038W	YNL307C	FYI
Kwal_7054	YAL030W	YBL050W	FYI
Kwal_7054	YAL030W	YDR468C	FYI
Kwal_7054	YAL030W	YPL232W	FYI
Kwal_7054	YOR327C	YOL018C	FYI
Kwal_7054	YOR327C, YAL030W	YGR009C	FYI
Kwal_7055	YAL029C	YBR130C	TAP
Kwal_7055	YAL029C	YFL039C	FYI
Kwal_7055	YAL029C	YKL130C	TAP
Kwal_7055	YOR326W	YBR109C	FYI,TAP
Kwal_7055	YOR326W	YIL070C	TAP
Kwal_7055	YOR326W	YOR035C	TAP
Kwal_7055	YOR326W, YAL029C	YGL106W	FYI,TAP
Kwal_7055	YOR326W, YAL029C	YHR023W	FYI,TAP
Kwal_7154	YAL017W	YDR099W	TAP
Kwal_7338	YJL099W	YLR330W	TAP
Kwal_7338	YJL099W	YMR116C	TAP
Kwal_7338	YJL099W	YMR237W	TAP
Kwal_7338	YJL099W	YOR299W	TAP
Kwal_7338	YKR027W	YGR161C	TAP
Kwal_7343	YJL098W	YER155C	TAP
Kwal_7343	YJL098W	YGL197W	TAP
Kwal_7343	YJL098W	YGR161C	TAP
Kwal_7343	YJL098W	YLR310C	TAP
Kwal_7343	YJL098W	YOR267C	TAP
Kwal_7343	YKR028W	YPL049C	TAP
Kwal_7343	YKR028W, YJL098W	YDL047W	FYI,TAP
Kwal_7462	YGL049C	YNL251C	TAP
Kwal_7462	YGL049C	YPL178W	TAP
Kwal_7462	YGR162W	YAL036C	TAP
Kwal_7462	YGR162W	YCR077C	TAP
Kwal_7462	YGR162W	YDL043C	FYI
Kwal_7462	YGR162W	YDL051W	TAP

Kwal_7462	YGR162W	YDL087C	TAP
Kwal_7462	YGR162W	YGR285C	TAP
Kwal_7462	YGR162W	YHL034C	TAP
Kwal_7462	YGR162W	YIL061C	TAP
Kwal_7462	YGR162W	YJL138C	FYI,TAP
Kwal_7462	YGR162W	YKR059W	FYI
Kwal_7462	YGR162W	YLR175W	TAP
Kwal_7462	YGR162W	YMR230W	TAP
Kwal_7462	YGR162W	YNL262W	TAP
Kwal_7462	YGR162W	YOR243C	TAP
Kwal_7462	YGR162W	YOR276W	FYI
Kwal_7462	YGR162W, YGL049C	YER165W	FYI
Kwal_7462	YGR162W, YGL049C	YIR001C	TAP
Kwal_7462	YGR162W, YGL049C	YMR125W	TAP
Kwal_7462	YGR162W, YGL049C	YOL139C	FYI,TAP
Kwal_7587	YMR109W	YBR109C	TAP
Kwal_7587	YMR109W	YBR177C	TAP
Kwal_7587	YMR109W	YDL019C	TAP
Kwal_7913	YHL001W, YKL006W	YER056C-A	FYI
Kwal_7913	YHL001W, YKL006W	YIL052C	FYI
Kwal_8043	YMR072W	YBR245C	TAP
Kwal_8043	YMR072W	YCR052W	TAP
Kwal_8043	YMR072W	YDR303C	TAP
Kwal_8043	YMR072W	YFR037C	TAP
Kwal_8043	YMR072W	YOL004W	TAP
Kwal_8043	YMR072W	YPL082C	TAP
Kwal_8387	YNL098C	YLL016W	FYI
Kwal_8387	YNL098C	YLR310C	FYI
Kwal_8387	YNL098C	YOL081W	FYI
Kwal_8433	YNL104C	YOR108W	TAP
Kwal_8433	YOR108W	YNL104C	TAP
Kwal_8703	YOR231W, YPL140C	YHR030C	FYI
Kwal_8703	YOR231W, YPL140C	YJL095W	FYI
Kwal_8703	YOR231W, YPL140C	YLL021W	FYI
Kwal_8703	YOR231W, YPL140C	YLR313C	FYI
Kwal_9752	YAR042W	YAR042W	TAP
Kwal_9752	YAR042W, YDL019C	YER120W	TAP
Kwal_9752	YDL019C	YMR109W	TAP
Kwal_10318	YML109W, YMR273C	YAL016W	TAP
Kwal_10318	YML109W, YMR273C	YDL188C	TAP
Kwal_10318	YML109W, YMR273C	YGL190C	TAP
Kwal_10393	YML100W	YMR261C	FYI
Kwal_10393	YML100W, YMR261C	YBR126C	FYI,TAP
Kwal_10393	YML100W, YMR261C	YDR074W	FYI,TAP
Kwal_10393	YMR261C	YML100W	FYI

Kwal_10573	YBL106C, YPR032W	YGR009C	FYI
Kwal_10573	YPR032W	YHR023W	FYI
Kwal_10720	YBL087C	YDR496C	TAP
Kwal_10720	YBL087C	YPL211W	TAP
Kwal_10720	YER117W	YOL077C	TAP
Kwal_10720	YER117W, YBL087C	YBL027W	FYI
Kwal_10720	YER117W, YBL087C	YBL092W	FYI
Kwal_10720	YER117W, YBL087C	YBR084C-A	FYI
Kwal_10720	YER117W, YBL087C	YDL136W	FYI
Kwal_10720	YER117W, YBL087C	YDL191W	FYI
Kwal_10720	YER117W, YBL087C	YGL103W	FYI
Kwal_10720	YER117W, YBL087C	YGR034W	FYI
Kwal_10720	YER117W, YBL087C	YIL018W	FYI
Kwal_10720	YER117W, YBL087C	YLR075W	FYI
Kwal_10720	YER117W, YBL087C	YLR344W	FYI
Kwal_10720	YER117W, YBL087C	YOL127W	FYI
Kwal_10720	YER117W, YBL087C	YOR063W	FYI
Kwal_10720	YER117W, YBL087C	YPL131W	FYI
Kwal_10733	YBL085W	YLR229C	FYI
Kwal_10733	YBL085W, YER114C	YBR200W	FYI
Kwal_10817	YBL075C, YER103W	YLR310C	FYI
Kwal_10827	YBL072C	YOR056C	TAP
Kwal_10827	YBL072C	YPL012W	TAP
Kwal_10827	YBL072C	YPR144C	TAP
Kwal_10827	YER102W	YLR192C	TAP
Kwal_10911	YER089C, YBL056W	YDR071C	TAP
Kwal_10991	YDR001C	YDR099W	TAP
Kwal_10991	YDR001C	YER177W	TAP
Kwal_11859	YDR480W, YPL049C	YBL016W	FYI
Kwal_11859	YDR480W, YPL049C	YGR040W	FYI
Kwal_11859	YDR480W, YPL049C	YHR084W	FYI
Kwal_11859	YPL049C	YKR028W	TAP
Kwal_12088	YBR189W	YDL060W	TAP
Kwal_12088	YBR189W	YGR081C	TAP
Kwal_12088	YBR189W	YNL132W	TAP
Kwal_12088	YBR189W	YOR056C	TAP
Kwal_12088	YPL081W	YBR079C	TAP
Kwal_12088	YPL081W	YNL207W	TAP
Kwal_12088	YPL081W	YPL204W	TAP
Kwal_12088	YPL081W, YBR189W	YNL178W	FYI
Kwal_12200	YBR177C	YMR109W	TAP
Kwal_12262	YPL106C	YGL206C	TAP
Kwal_12262	YPL106C	YML028W	TAP
Kwal_12461	YBL027W, YBR084C-A	YBL087C	FYI
Kwal_12461	YBL027W, YBR084C-A	YBL092W	FYI

Kwal_12461	YBL027W, YBR084C-A	YBR031W	FYI
Kwal_12461	YBL027W, YBR084C-A	YDL136W	FYI
Kwal_12461	YBL027W, YBR084C-A	YDL191W	FYI
Kwal_12461	YBL027W, YBR084C-A	YDR012W	FYI
Kwal_12461	YBL027W, YBR084C-A	YER117W	FYI
Kwal_12461	YBL027W, YBR084C-A	YGL103W	FYI
Kwal_12461	YBL027W, YBR084C-A	YGL147C	FYI
Kwal_12461	YBL027W, YBR084C-A	YGR034W	FYI
Kwal_12461	YBL027W, YBR084C-A	YGR085C	FYI
Kwal_12461	YBL027W, YBR084C-A	YIL018W	FYI
Kwal_12461	YBL027W, YBR084C-A	YJL177W	FYI
Kwal_12461	YBL027W, YBR084C-A	YKL180W	FYI
Kwal_12461	YBL027W, YBR084C-A	YLR344W	FYI
Kwal_12461	YBL027W, YBR084C-A	YNL067W	FYI
Kwal_12461	YBL027W, YBR084C-A	YOL127W	FYI
Kwal_12461	YBL027W, YBR084C-A	YOR063W	FYI
Kwal_12461	YBL027W, YBR084C-A	YPL131W	FYI
Kwal_12461	YBL027W, YBR084C-A	YPR102C	FYI
Kwal_12655	YJL110C	YKR034W	FYI
Kwal_12655	YKR034W	YJL110C	FYI
Kwal_12693	YBR031W, YDR012W	YBL027W	FYI
Kwal_12693	YBR031W, YDR012W	YBL092W	FYI
Kwal_12693	YBR031W, YDR012W	YBR084C-A	FYI
Kwal_12693	YBR031W, YDR012W	YDL136W	FYI
Kwal_12693	YBR031W, YDR012W	YDL191W	FYI
Kwal_12693	YBR031W, YDR012W	YGL103W	FYI
Kwal_12693	YBR031W, YDR012W	YGR034W	FYI
Kwal_12693	YBR031W, YDR012W	YIL018W	FYI
Kwal_12693	YBR031W, YDR012W	YLR075W	FYI
Kwal_12693	YBR031W, YDR012W	YLR344W	FYI
Kwal_12693	YBR031W, YDR012W	YOL127W	FYI
Kwal_12693	YBR031W, YDR012W	YOR063W	FYI
Kwal_12693	YBR031W, YDR012W	YPL131W	FYI
Kwal_12745	YHR152W	YDR267C	FYI
Kwal_12745	YHR152W	YER032W	FYI
Kwal_12745	YHR152W	YHR073W	FYI
Kwal_12745	YHR152W	YHR128W	FYI
Kwal_12745	YHR152W	YJL168C	FYI
Kwal_12745	YHR152W	YJR148W	FYI
Kwal_12745	YHR152W	YLR016C	FYI
Kwal_12745	YHR152W	YLR132C	FYI
Kwal_12745	YHR152W	YLR288C	FYI
Kwal_12745	YHR152W	YPR118W	FYI
Kwal_12745	YHR152W	YPR152C	FYI
Kwal_13222	YCL024W	YDR507C	TAP

Kwal_13222	YDR507C	YCL024W	TAP
Kwal_13222	YDR507C	YCR002C	FYI
Kwal_13222	YDR507C	YDL225W	FYI
Kwal_13222	YDR507C	YHR107C	FYI
Kwal_13222	YDR507C	YJR076C	FYI
Kwal_13222	YDR507C	YLR314C	FYI
Kwal_13222	YDR507C, YCL024W	YKR048C	FYI
Kwal_13256	YDR502C	YLR153C	TAP
Kwal_13256	YDR502C	YLR180W	TAP
Kwal_13256	YLR180W	YDR502C	TAP
Kwal_13256	YLR180W	YER090W	TAP
Kwal_13256	YLR180W	YGL195W	TAP
Kwal_13676	YCR031C, YJL191W	YGL123W	FYI
Kwal_13676	YCR031C, YJL191W	YHL015W	FYI
Kwal_13676	YCR031C, YJL191W	YJR123W	FYI
Kwal_13676	YCR031C, YJL191W	YNL178W	FYI
Kwal_13676	YCR031C, YJL191W	YOL040C	FYI
Kwal_13736	YKL180W, YJL177W	YBL027W	FYI
Kwal_13736	YKL180W, YJL177W	YBL092W	FYI
Kwal_13736	YKL180W, YJL177W	YBR084C-A	FYI
Kwal_13736	YKL180W, YJL177W	YDL136W	FYI
Kwal_13736	YKL180W, YJL177W	YDL191W	FYI
Kwal_13736	YKL180W, YJL177W	YGL103W	FYI
Kwal_13736	YKL180W, YJL177W	YGR034W	FYI
Kwal_13736	YKL180W, YJL177W	YIL018W	FYI
Kwal_13736	YKL180W, YJL177W	YLR075W	FYI
Kwal_13736	YKL180W, YJL177W	YLR344W	FYI
Kwal_13736	YKL180W, YJL177W	YOL127W	FYI
Kwal_13736	YKL180W, YJL177W	YOR063W	FYI
Kwal_13736	YKL180W, YJL177W	YPL131W	FYI
Kwal_13846	YJL164C	YKL166C	FYI
Kwal_13846	YJL164C, YKL166C	YIL033C	FYI
Kwal_13846	YJL164C, YKL166C	YPL203W	FYI
Kwal_13846	YKL166C	YJL164C	FYI
Kwal_14000	YJL138C	YKR059W	TAP
Kwal_14000	YKR059W	YJL138C	TAP
Kwal_14000	YKR059W, YJL138C	YGR162W	FYI,TAP
Kwal_14267	YML085C	YCL029C	FYI
Kwal_14267	YML085C	YFL037W	FYI
Kwal_14267	YML085C	YNL223W	FYI
Kwal_14267	YML085C	YOL086C	FYI
Kwal_14267	YML124C	YER016W	FYI
Kwal_14353	YDR309C	YER149C	FYI
Kwal_14353	YDR309C	YLR319C	FYI
Kwal_14353	YHR061C, YDR309C	YLR229C	FYI

Kwal_14596	YHR135C	YER123W	FYI
Kwal_14596	YHR135C	YNL154C	FYI
Kwal_14596	YNL154C	YHR135C	FYI
Kwal_14899	YHR161C, YGR241C	YGL206C	FYI
Kwal_14899	YHR161C, YGR241C	YGR167W	FYI
Kwal_14899	YHR161C, YGR241C	YIR006C	FYI
Kwal_14989	YKL068W	YMR047C	FYI
Kwal_14989	YMR047C	YER107C	FYI
Kwal_14989	YMR047C	YJL061W	FYI
Kwal_14989	YMR047C	YKL068W	FYI
Kwal_14989	YMR047C, YKL068W	YLR347C	FYI
Kwal_14989	YMR047C, YKL068W	YPL169C	FYI
Kwal_15007	YGR180C	YJL026W	FYI,TAP
Kwal_15007	YGR180C, YJL026W	YER070W	FYI
Kwal_15007	YGR180C, YJL026W	YIL066C	FYI
Kwal_15007	YJL026W	YGR180C	FYI,TAP
Kwal_15889	YOR312C, YMR242C	YDL075W	FYI
Kwal_15889	YOR312C, YMR242C	YGL030W	FYI
Kwal_15889	YOR312C, YMR242C	YJL189W	FYI
Kwal_15889	YOR312C, YMR242C	YLR406C	FYI
Kwal_16748	YHR203C, YJR145C	YBR048W	FYI
Kwal_16748	YHR203C, YJR145C	YDL061C	FYI
Kwal_16748	YHR203C, YJR145C	YDR025W	FYI
Kwal_16748	YHR203C, YJR145C	YGL123W	FYI
Kwal_16748	YHR203C, YJR145C	YJL190C	FYI
Kwal_16748	YHR203C, YJR145C	YLR367W	FYI
Kwal_16748	YHR203C, YJR145C	YLR388W	FYI
Kwal_16748	YHR203C, YJR145C	YNL178W	FYI
Kwal_16748	YHR203C, YJR145C	YOL040C	FYI
Kwal_16781	YHR208W	YJR148W	TAP
Kwal_16781	YHR208W	YLR259C	TAP
Kwal_16781	YJR148W	YHR152W	FYI
Kwal_16781	YJR148W	YHR208W	TAP
Kwal_16977	YLR442C	YDR227W	FYI
Kwal_16977	YLR442C	YNL216W	FYI
Kwal_16977	YML065W	YBR060C	FYI,TAP
Kwal_16977	YML065W	YHR118C	FYI,TAP
Kwal_16977	YML065W	YKR101W	FYI
Kwal_16977	YML065W	YLL004W	FYI,TAP
Kwal_16977	YML065W	YNL261W	FYI,TAP
Kwal_16977	YML065W	YPR162C	FYI,TAP
Kwal_16988	YLR441C, YML063W	YDR064W	FYI
Kwal_17049	YJR009C	YDL188C	TAP
Kwal_17056	YML058W	YER070W	FYI
Kwal_17070	YLR433C, YML057W	YBR109C	FYI

Kwal_17070	YLR433C, YML057W	YKL190W	FYI
Kwal_17202	YAL007C	YAR002C-A	FYI,TAP
Kwal_17202	YAL007C	YGL200C	FYI,TAP
Kwal_17202	YAL007C	YML012W	FYI,TAP
Kwal_17263	YER129W	YDR422C	TAP
Kwal_17263	YER129W	YDR477W	TAP
Kwal_17263	YER129W	YER027C	TAP
Kwal_17263	YER129W	YGL115W	TAP
Kwal_17576	YAR002C-A	YAL007C	FYI,TAP
Kwal_17576	YAR002C-A	YGL200C	FYI,TAP
Kwal_17576	YAR002C-A	YML012W	FYI,TAP
Kwal_17726	YLR028C	YCR073W-A	TAP
Kwal_17726	YLR028C	YER133W	TAP
Kwal_17726	YLR028C	YER177W	TAP
Kwal_17733	YLR029C, YMR121C	YPR043W	FYI
Kwal_18059	YGR056W	YHR056C	TAP
Kwal_18059	YGR056W	YMR033W	TAP
Kwal_18059	YLR357W	YBR049C	TAP
Kwal_18059	YLR357W	YBR245C	TAP
Kwal_18059	YLR357W	YDR224C	TAP
Kwal_18059	YLR357W	YDR225W	TAP
Kwal_18059	YLR357W	YGR275W	TAP
Kwal_18059	YLR357W, YGR056W	YCR052W	TAP
Kwal_18059	YLR357W, YGR056W	YDR303C	TAP
Kwal_18059	YLR357W, YGR056W	YFR037C	TAP
Kwal_18059	YLR357W, YGR056W	YIL126W	TAP
Kwal_18059	YLR357W, YGR056W	YKR008W	TAP
Kwal_18059	YLR357W, YGR056W	YLR033W	TAP
Kwal_18059	YLR357W, YGR056W	YLR321C	TAP
Kwal_18059	YLR357W, YGR056W	YML127W	TAP
Kwal_18059	YLR357W, YGR056W	YMR091C	TAP
Kwal_18059	YLR357W, YGR056W	YPR034W	TAP
Kwal_18439	YML028W	YPL106C	TAP
Kwal_18456	YDR450W, YML026C	YGL123W	FYI
Kwal_18456	YDR450W, YML026C	YHL015W	FYI
Kwal_18456	YDR450W, YML026C	YJR123W	FYI
Kwal_18456	YDR450W, YML026C	YNL178W	FYI
Kwal_18456	YDR450W, YML026C	YOL040C	FYI
Kwal_18631	YML007W	YBR081C	TAP
Kwal_18631	YML007W	YBR216C	FYI
Kwal_18631	YML007W	YBR253W	TAP
Kwal_18631	YML007W	YBR289W	TAP
Kwal_18631	YML007W	YCL010C	TAP
Kwal_18631	YML007W	YDR359C	TAP
Kwal_18631	YML007W	YDR448W	TAP

Kwal_18631	YML007W	YER110C	FYI
Kwal_18631	YML007W	YFL024C	TAP
Kwal_18631	YML007W	YGL112C	TAP
Kwal_18631	YML007W	YGL151W	TAP
Kwal_18631	YML007W	YGR218W	FYI
Kwal_18631	YML007W	YHR041C	TAP
Kwal_18631	YML007W	YHR058C	TAP
Kwal_18631	YML007W	YJL081C	TAP
Kwal_18631	YML007W	YJL176C	TAP
Kwal_18631	YML007W	YNL236W	TAP
Kwal_18631	YML007W	YNR023W	TAP
Kwal_18631	YML007W	YOL148C	TAP
Kwal_18631	YML007W	YOR244W	TAP
Kwal_18631	YML007W	YOR290C	TAP
Kwal_18631	YML007W	YPL254W	TAP
Kwal_18631	YML007W	YPR070W	TAP
Kwal_18673	YDR418W	YAL035W	TAP
Kwal_18673	YDR418W	YGR285C	TAP
Kwal_18673	YDR418W	YHR064C	TAP
Kwal_18673	YEL054C	YGL099W	TAP
Kwal_18673	YEL054C, YDR418W	YDL136W	FYI
Kwal_18673	YEL054C, YDR418W	YDL191W	FYI
Kwal_18673	YEL054C, YDR418W	YGL103W	FYI
Kwal_18673	YEL054C, YDR418W	YGR034W	FYI
Kwal_18673	YEL054C, YDR418W	YIL018W	FYI
Kwal_18673	YEL054C, YDR418W	YLR075W	FYI
Kwal_18673	YEL054C, YDR418W	YLR344W	FYI
Kwal_18673	YEL054C, YDR418W	YOL127W	FYI
Kwal_18673	YEL054C, YDR418W	YOR063W	FYI
Kwal_18673	YEL054C, YDR418W	YPL131W	FYI
Kwal_19040	YER027C	YDR422C	TAP
Kwal_19040	YER027C	YER129W	TAP
Kwal_19040	YER027C	YGL208W	FYI
Kwal_19040	YGL208W	YER027C	FYI
Kwal_19040	YGL208W, YER027C	YDR477W	FYI,TAP
Kwal_19040	YGL208W, YER027C	YGL115W	FYI,TAP
Kwal_19987	YDR264C	YAL041W	FYI
Kwal_19987	YDR264C	YDL226C	FYI
Kwal_19987	YDR264C	YDR103W	FYI
Kwal_19987	YDR264C	YJR086W	FYI
Kwal_19987	YDR264C	YOR212W	FYI
Kwal_19987	YDR264C	YPL242C	FYI
Kwal_20268	YMR237W, YOR299W	YJL099W	TAP
Kwal_20268	YMR237W, YOR299W	YLR330W	TAP
Kwal_20419	YFR031C-A	YAL035W	TAP

Kwal_20419	YFR031C-A	YDL082W	TAP
Kwal_20419	YFR031C-A	YNL301C	TAP
Kwal_20419	YFR031C-A, YIL018W	YGR085C	FYI,TAP
Kwal_20419	YIL018W	YBL027W	FYI
Kwal_20419	YIL018W	YBL087C	FYI
Kwal_20419	YIL018W	YBL092W	FYI
Kwal_20419	YIL018W	YBR031W	FYI
Kwal_20419	YIL018W	YBR084C-A	FYI
Kwal_20419	YIL018W	YDL136W	FYI
Kwal_20419	YIL018W	YDL191W	FYI
Kwal_20419	YIL018W	YDR012W	FYI
Kwal_20419	YIL018W	YDR418W	FYI
Kwal_20419	YIL018W	YEL054C	FYI
Kwal_20419	YIL018W	YER117W	FYI
Kwal_20419	YIL018W	YGL103W	FYI
Kwal_20419	YIL018W	YGL135W	FYI
Kwal_20419	YIL018W	YGL147C	FYI
Kwal_20419	YIL018W	YGR034W	FYI
Kwal_20419	YIL018W	YIL133C	FYI
Kwal_20419	YIL018W	YJL177W	FYI
Kwal_20419	YIL018W	YKL180W	FYI
Kwal_20419	YIL018W	YLR075W	FYI
Kwal_20419	YIL018W	YLR340W	FYI
Kwal_20419	YIL018W	YLR344W	FYI
Kwal_20419	YIL018W	YNL067W	FYI
Kwal_20419	YIL018W	YNL069C	FYI
Kwal_20419	YIL018W	YOL127W	FYI
Kwal_20419	YIL018W	YOR063W	FYI
Kwal_20419	YIL018W	YPL131W	FYI
Kwal_20419	YIL018W	YPL220W	FYI
Kwal_20419	YIL018W	YPR102C	FYI
Kwal_20474	YPR120C	YLR079W	FYI
Kwal_20474	YPR120C, YGR109C	YBR160W	FYI
Kwal_20479	YPR119W	YBR135W	FYI
Kwal_20479	YPR119W	YDR025W	FYI
Kwal_20479	YPR119W	YER111C	FYI
Kwal_20479	YPR119W	YKR048C	FYI
Kwal_20479	YPR119W, YGR108W	YBR160W	FYI
Kwal_20547	YGR097W, YPR115W	YDR167W	FYI
Kwal_20756	YER132C	YDR103W	FYI
Kwal_20756	YGL197W	YDL047W	TAP
Kwal_20756	YGL197W	YJL098W	TAP
Kwal_20756	YGL197W	YLR310C	TAP
Kwal_20756	YGL197W	YMR117C	FYI
Kwal_21903	YPR162C	YBR060C	FYI,TAP

Kwal_21903	YPR162C	YHR118C	TAP
Kwal_21903	YPR162C	YML065W	FYI,TAP
Kwal_21903	YPR162C	YNL261W	TAP
Kwal_22001	YNL307C	YAL038W	FYI
Kwal_22001	YNL307C	YGR140W	FYI
Kwal_22001	YNL307C	YLR175W	FYI
Kwal_22853	YDL042C	YDR227W	FYI
Kwal_22853	YDL042C	YJL076W	FYI
Kwal_22992	YBR048W, YDR025W	YGL123W	FYI
Kwal_22992	YBR048W, YDR025W	YHL015W	FYI
Kwal_22992	YBR048W, YDR025W	YHR203C	FYI
Kwal_22992	YBR048W, YDR025W	YJR123W	FYI
Kwal_22992	YBR048W, YDR025W	YJR145C	FYI
Kwal_22992	YBR048W, YDR025W	YNL178W	FYI
Kwal_22992	YBR048W, YDR025W	YOL040C	FYI
Kwal_22992	YDR025W	YPR119W	FYI
Kwal_22993	YBR049C	YDR303C	TAP
Kwal_22993	YBR049C	YLR357W	TAP
Kwal_22993	YBR049C	YPL082C	TAP
Kwal_22993	YBR049C	YPR110C	TAP
Kwal_23042	YBR052C	YDR032C	TAP
Kwal_23042	YDR032C	YBR052C	TAP
Kwal_23144	YOL081W	YNL098C	FYI
Kwal_23198	YOL086C	YML085C	FYI
Kwal_23262	YLR273C	YLR258W	FYI
Kwal_23262	YOR178C	YBR045C	FYI
Kwal_23294	YLR277C	YDR228C	TAP
Kwal_23294	YLR277C	YDR301W	TAP
Kwal_23294	YLR277C	YJR093C	TAP
Kwal_23294	YLR277C	YKL059C	TAP
Kwal_23294	YLR277C	YKR002W	TAP
Kwal_23294	YLR277C	YLR115W	TAP
Kwal_23294	YLR277C	YNL317W	TAP
Kwal_23294	YLR277C	YPR107C	TAP
Kwal_23294	YOR179C	YGR156W	TAP
Kwal_23294	YOR179C, YLR277C	YAL043C	TAP
Kwal_23294	YOR179C, YLR277C	YDR195W	TAP
Kwal_23294	YOR179C, YLR277C	YER133W	TAP
Kwal_23294	YOR179C, YLR277C	YKL018W	TAP
Kwal_23506	YDR098C, YER174C	YGL220W	FYI
Kwal_23522	YDR099W	YAL017W	TAP
Kwal_23522	YDR099W	YER177W	TAP
Kwal_23522	YER177W	YDR099W	TAP
Kwal_23522	YER177W	YER133W	TAP
Kwal_23522	YER177W	YLR028C	TAP

Kwal_23522	YER177W, YDR099W	YDR001C	TAP
Kwal_23522	YER177W, YDR099W	YGL252C	TAP
Kwal_24014	YGL076C, YPL198W	YBL092W	FYI
Kwal_24014	YGL076C, YPL198W	YOL127W	FYI
Kwal_24014	YGL076C, YPL198W	YPL131W	FYI
Kwal_24134	YGL133W	YBR245C	TAP
Kwal_24134	YGL133W	YDR121W	TAP
Kwal_24134	YGL133W	YDR224C	TAP
Kwal_24134	YGL133W	YFR013W	TAP
Kwal_24134	YGL133W	YOR304W	TAP
Kwal_24151	YGL134W, YPL219W	YPL031C	FYI
Kwal_24153	YGL135W, YPL220W	YDL136W	FYI
Kwal_24153	YGL135W, YPL220W	YDL191W	FYI
Kwal_24153	YGL135W, YPL220W	YGL103W	FYI
Kwal_24153	YGL135W, YPL220W	YGR034W	FYI
Kwal_24153	YGL135W, YPL220W	YIL018W	FYI
Kwal_24153	YGL135W, YPL220W	YLR075W	FYI
Kwal_24153	YGL135W, YPL220W	YLR344W	FYI
Kwal_24153	YGL135W, YPL220W	YOL127W	FYI
Kwal_24153	YGL135W, YPL220W	YOR063W	FYI
Kwal_24153	YGL135W, YPL220W	YPL131W	FYI
Kwal_24259	YPL232W	YAL030W	FYI
Kwal_24259	YPL232W, YMR183C	YGR009C	FYI
Kwal_24289	YMR186W	YHR102W	FYI
Kwal_24289	YPL240C	YAL005C	FYI
Kwal_24289	YPL240C	YKL117W	FYI
Kwal_24289	YPL240C	YLR216C	FYI
Kwal_24289	YPL240C	YLR310C	FYI
Kwal_24289	YPL240C	YOR027W	FYI
Kwal_24289	YPL240C, YMR186W	YBR155W	FYI
Kwal_24289	YPL240C, YMR186W	YLR362W	FYI
Kwal_24375	YMR192W	YNL112W	TAP
Kwal_24375	YMR192W	YPL249C	TAP
Kwal_24375	YPL249C	YMR192W	TAP
Kwal_24388	YMR194W	YGL099W	TAP
Kwal_24388	YMR194W	YNL132W	TAP
Kwal_24388	YPL249C-A	YBR142W	TAP
Kwal_24388	YPL249C-A	YDR101C	TAP
Kwal_24388	YPL249C-A	YNL301C	TAP
Kwal_24430	YDL188C	YJR009C	TAP
Kwal_24430	YDL188C	YML109W	TAP
Kwal_24430	YDL188C	YMR273C	TAP
Kwal_24430	YDL188C, YDL134C	YAL016W	FYI,TAP
Kwal_24430	YDL188C, YDL134C	YGL190C	TAP
Kwal_24430	YDL188C, YDL134C	YMR028W	FYI

Kwal_24430	YDL188C, YDL134C	YOR014W	FYI,TAP
Kwal_24561	YDL155W	YBR135W	FYI
Kwal_24561	YLR210W, YDL155W	YBR160W	FYI

Bibliography

[Ad06] Alexandre H. Abdo and A. P. S. de Moura, *Measuring the local topology of networks: an extended clustering coefficient*, 2006.

[Art91] Michael Artin, *Algebra*, Prentice Hall Inc., Englewood Cliffs, NJ, 1991.

[Bar02] Albert-László Barabási, *Linked*, Perseus Publishing, Cambridge, MA, 2002.

[Bar03] Albert-László Barabási, *Emergence of scaling in complex networks*, Handbook of graphs and networks, Wiley-VCH, Weinheim, 2003, pp. 69–84.

[BCD+04] Alex Bateman, Lachlan Coin, Richard Durbin, Robert D Finn, Volker Hollich, Sam Griffiths-Jones, Ajay Khanna, Mhairi Marshall, Simon Moxon, Erik L L Sonnhammer, David J Studholme, Corin Yeats, and Sean R Eddy, *The pfam protein families database*, Nucleic Acids Res **32** (2004), no. Database issue, 138–141.

[BdW05] G. Bounova and O. L. de Weck, *Graph-theoretical considerations in design of large telescope arrays for robustness and scalability*, AIAA Multidisciplinary Design Optimization Specialist Conference, 2005, p. 2063.

[BLW04] Johannes Berg, Michael Lassig, and Andreas Wagner, *Structure and evolution of protein interaction networks: a statistical model for link dynamics and gene duplications*, BMC Evol Biol **4** (2004), no. 1, 51.

[BP06] Kim Baskerville and Maya Paczuski, *Subgraph ensembles and motif discovery using a new heuristic for graph isomorphism*, 2006, arxiv.org:q-bio/0606023.

[CBN90] Joel E. Cohen, Frédéric Briand, and Charles M. Newman, *Community food webs*, Biomathematics, vol. 20, Springer-Verlag, Berlin, 1990, Data and theory, With a contribution by Zbigniew J. Palka.

[CL86] C Chothia and A M Lesk, *The relation between the divergence of sequence and structure in proteins*, EMBO J **5** (1986), no. 4, 823–826.

[CLDG03] Fan Chung, Linyuan Lu, T Gregory Dewey, and David J Galas, *Duplication models for biological networks*, J Comput Biol **10** (2003), no. 5, 677–687.

[DMSC02] Minghua Deng, Shipra Mehta, Fengzhu Sun, and Ting Chen, *Inferring domain-domain interactions from protein-protein interactions*, Genome Res **12** (2002), no. 10, 1540–1548.

[ER59] P. Erdős and A. Rényi, *On random graphs. I*, Publ. Math. Debrecen **6** (1959), 290–297.

[FNS+06] J. Flannick, A. Novak, B. S. Srinivasan, H. H. McAdams, and S. Batzoglou, *Graemlin: general and robust alignment of multiple large interaction networks*, submitted, 2006.

[GAG+06] Anne-Claude Gavin, Patrick Aloy, Paola Grandi, Roland Krause, Markus Boesche, Martina Marzioch, Christina Rau, Lars Juhl Jensen, Sonja Bastuck, Birgit Dumpelfeld, Angela Edelmann, Marie-Anne Heurtier, Verena Hoffman, Christian Hoefert, Karin Klein, Manuela Hudak, Anne-Marie Michon, Malgorzata Schelder, Markus Schirle, Marita Remor, Tatjana Rudi, Sean Hooper, Andreas Bauer, Tewis Bouwmeester, Georg Casari, Gerard Drewes, Gitte Neubauer, Jens M Rick, Bernhard Kuster, Peer Bork, Robert B Russell, and Giulio

Superti-Furga, *Proteome survey reveals modularity of the yeast cell machinery*, Nature **440** (2006), no. 7084, 631–636.

[Gla02] Malcolm Gladwell, *The tipping point: How little things can make a big difference*, Little, Brown, and Company, New York, NY, 2002.

[GMZ03] C. Gkantsidis, M. Mihail, and E. Zegura, *The markov chain simulation method for generating connected power law random graphs*, SIAM Proc. 5th Workshop on Algorithm Engineering and Experiments (ALENEX), 2003.

[GSW04] Michael A Gilchrist, Laura A Salter, and Andreas Wagner, *A statistical framework for combining and interpreting proteomic datasets*, Bioinformatics **20** (2004), no. 5, 689–700, Evaluation Studies.

[Hal04] Jennifer Hallinan, *Gene duplication and hierarchical modularity in intracellular interaction networks*, Biosystems **74** (2004), no. 1-3, 51–62, Evaluation Studies.

[HBH+04] Jing-Dong J Han, Nicolas Bertin, Tong Hao, Debra S Goldberg, Gabriel F Berriz, Lan V Zhang, Denis Dupuy, Albertha J M Walhout, Michael E Cusick, Frederick P Roth, and Marc Vidal, *Evidence for dynamically organized modularity in the yeast protein-protein interaction network*, Nature **430** (2004), no. 6995, 88–93.

[HEO05] D. F. Holt, B. Eick, and E. A. O'Brien, *Handbook of computational group theory*, Chapman & Hall/CRC, London, UK, 2005.

[HRO05] L. Hakes, D. L. Robertson, and S. G. Oliver, *Effect of dataset selection on the topological interpretation of protein interaction networks*, BMC Genomics **6** (2005), 131–138.

[HV03] R. "Hoffman and A." Valencia, *Protein interaction: same network, different hubs*, Trends in Genetics **19** (2003), 681–683.

[IKMY05] I. Ispolatov, P.L. Krapivsky, I. Mazo, and A. Yuryev, *Cliques and duplication-divergence network growth*, New J. Phys. **7** (2005), 145.

[IMD06] *The internet movie database (imdb)*, online, 2006, http://www.imdb.com/.

[KBL04] Manolis Kellis, Bruce W Birren, and Eric S Lander, *Proof and evolutionary analysis of ancient genome duplication in the yeast Saccharomyces cerevisiae*, Nature **428** (2004), no. 6983, 617–624.

[KBL06] P. D. Kuo, W. Banzhaf, and A Leier, *Network topology and the evolution of dynamics in an artificial genetic regulatory network model created by whole genome duplication and divergence*, BioSystems (2006), in press.

[KGG06] Mathias Kuhnt, Ingmar Glauche, and Martin Greiner, *Impact of observational incompleteness on the structural properties of protein interaction networks*, 2006, arxiv.org:q-bio/0605033, submitted to Elsevier Science.

[KIMA04a] N Kashtan, S Itzkovitz, R Milo, and U Alon, *Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs*, Bioinformatics **20** (2004), no. 11, 1746–1758, Evaluation Studies.

[KIMA04b] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon, *Topological generalizations of network motifs*, Phys. Rev. E **70** (2004), 031909.

[KMP+01] S Kalir, J McClure, K Pabbaraju, C Southward, M Ronen, S Leibler, M G Surette, and U Alon, *Ordering genes in a flagella pathway by analysis of expression kinetics from living bacteria*, Science **292** (2001), no. 5524, 2080–2083.

[KSK+03] Brian P Kelley, Roded Sharan, Richard M Karp, Taylor Sittler, David E Root, Brent R Stockwell, and Trey Ideker, *Conserved pathways within*

bacteria and yeast as revealed by global protein network alignment, Proc Natl Acad Sci U S A **100** (2003), no. 20, 11394–11399.

[LY00] T. I. Lee and R. A. Young, *Transcription of eukaryotic protein-coding genes*, Annu. Rev. Genet. **34** (2000), 77–137.

[MA03] S Mangan and U Alon, *Structure and function of the feed-forward loop network motif*, Proc Natl Acad Sci U S A **100** (2003), no. 21, 11980–11985.

[MBV03] Thomas Manke, Ricardo Bringas, and Martin Vingron, *Correlating protein-DNA and protein-protein interaction networks*, J Mol Biol **333** (2003), no. 1, 75–85.

[McK81] Brendan D. McKay, *Practical graph isomorphism*, Proceedings of the Tenth Manitoba Conference on Numerical Mathematics and Computing, Vol. I (Winnipeg, Man., 1980), vol. 30, 1981, pp. 45–87.

[MIK+04] Ron Milo, Shalev Itzkovitz, Nadav Kashtan, Reuven Levitt, Shai Shen-Orr, Inbal Ayzenshtat, Michal Sheffer, and Uri Alon, *Superfamilies of evolved and designed networks*, Science **303** (2004), no. 5663, 1538–1542.

[MKFV06] Priya Mahadevan, Dmitri Krioukov, Kevin Fall, and Amin Vahdat, *Systematic topology analysis and generation using degree correlations*, ACM SIGCOMM (2006), arXiv.org:cs/0605007.

[MSOI+02] R Milo, S Shen-Orr, S Itzkovitz, N Kashtan, D Chklovskii, and U Alon, *Network motifs: simple building blocks of complex networks*, Science **298** (2002), no. 5594, 824–827.

[MZA03] S Mangan, A Zaslaver, and U Alon, *The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks*, J Mol Biol **334** (2003), no. 2, 197–204.

[MZW05] Manuel Middendorf, Etay Ziv, and Chris H Wiggins, *Inferring network mechanisms: the Drosophila melanogaster protein interaction network*, Proc Natl Acad Sci U S A **102** (2005), no. 9, 3192–3197.

[Ohn70] S. Ohno, *Evolution by gene duplication*, Springer, Berlin, 1970.

[PEKK06] A. Presser, M. Elowitz, M. Kellis, and R. Kishony, Unpublished data, 2006.

[PSSS03] Romualdo Pastor-Satorras, Eric Smith, and Ricard V Sole, *Evolving protein interaction networks through gene duplication*, J Theor Biol **222** (2003), no. 2, 199–210.

[RK06] Matt Rasmussen and Manolis Kellis, in preparation, 2006.

[RRSA02] Michal Ronen, Revital Rosenberg, Boris I Shraiman, and Uri Alon, *Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics*, Proc Natl Acad Sci U S A **99** (2002), no. 16, 10555–10560.

[RS04] David J Reiss and Benno Schwikowski, *Predicting protein-peptide interactions via a network-based motif sampler*, Bioinformatics **20 Suppl 1** (2004), I274–I282.

[RSM+02] E Ravasz, A L Somera, D A Mongru, Z N Oltvai, and A L Barabasi, *Hierarchical organization of modularity in metabolic networks*, Science **297** (2002), no. 5586, 1551–1555.

[Ser03] Ákos Seress, *Permutation group algorithms*, Cambridge Tracts in Mathematics, vol. 152, Cambridge University Press, Cambridge, 2003.

[SOMMA02] Shai S Shen-Orr, Ron Milo, Shmoolik Mangan, and Uri Alon, *Network motifs in the transcriptional regulation network of Escherichia coli*, Nat Genet **31** (2002), no. 1, 64–68.

[Str03] S. Strogatz, *Sync: The emerging science of spontaneous order*, Hyperion Books, New York,NY, 2003.

[SV06] Ricard V Sole and Sergi Valverde, *Are network motifs the spandrels of cellular complexity?*, Trends Ecol Evol **21** (2006), no. 8, 419–422.

[THG94] J D Thompson, D G Higgins, and T J Gibson, *CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice*, Nucleic Acids Res **22** (1994), no. 22, 4673–4680.

[VDS+04] A Vazquez, R Dobrin, D Sergi, J-P Eckmann, Z N Oltvai, and A-L Barabasi, *The topological relationship between the large-scale attributes and local interaction patterns of complex networks*, Proc Natl Acad Sci U S A **101** (2004), no. 52, 17940–17945.

[Wag02] Andreas Wagner, *Asymmetric functional divergence of duplicate genes in yeast*, Mol Biol Evol **19** (2002), no. 10, 1760–1768.

[Wag03] Andreas Wagner, *How the global structure of protein interaction networks evolves*, Proc Biol Sci **270** (2003), no. 1514, 457–466.

[Wat99] D. J. Watts, *Small worlds: The dynamics of networks between order and randomness*, Princeton University Press, Princeton, NJ, 1999.

[Wat03] D. J. Watts, *Six degrees: The science of a connected age*, W. W. Norton & Company, Inc., New York, NY, 2003.

[WG05] Shiquan Wu and Xun Gu, *Gene network: model, dynamics and simulation*, Computing and combinatorics, Lecture Notes in Comput. Sci., vol. 3595, Springer, Berlin, 2005, pp. 12–21.

[Wil98] E. O. Wilson, *Consilience: The unity of knowledge*, Alfred A. Knopf, Inc., New York,NY, 1998.

[WM00] R J Williams and N D Martinez, *Simple rules yield complex food webs*, Nature **404** (2000), no. 6774, 180–183.

[WS98] D J Watts and S H Strogatz, *Collective dynamics of 'small-world' networks*, Nature **393** (1998), no. 6684, 440–442.

[ZLKK05] Ze Zhang, Z W Luo, Hirohisa Kishino, and Mike J Kearsey, *Divergence pattern of duplicate genes in protein-protein interactions follows the power law*, Mol Biol Evol **22** (2005), no. 3, 501–505.

[ZMR+04] Alon Zaslaver, Avi E Mayo, Revital Rosenberg, Pnina Bashkin, Hila Sberro, Miri Tsalyuk, Michael G Surette, and Uri Alon, *Just-in-time transcription program in metabolic pathways*, Nat Genet **36** (2004), no. 5, 486–491.