Graph Theory Approaches to Protein Interaction Data Analysis

Nataša Pržulj

September 8, 2003

Contents

1 Introduction

Understanding protein interactions is one of the important problems of computational biology. It is widely believed that studying networks of these interactions will provide valuable insight about the inner workings of cells leading, for example, to important insights into disease. These protein-protein interaction (PPI) networks are commonly represented in a graph format, with nodes corresponding to proteins and edges corresponding to protein-protein interactions. An example of a PPI network constructed in this way is presented in Figure 1. The volume of experimental data on protein-protein interactions is rapidly increasing thanks to high-throughput techniques which are able to produce large batches of PPIs. For example, yeast contains over 6,000 proteins, and currently over 78,000 PPIs have been identified between the yeast proteins, with hundreds of labs around the world adding to this list constantly. The analogous networks for mammals are expected to be much larger. For example, humans are expected to have around $120,000$ proteins and around 10^6 PPIs. The relationships between the structure of a PPI network and a cellular function are just starting to be explored. Comparisons between PPI networks of different organisms may reveal a common structure and offer an explanation of why natural selection favored it.

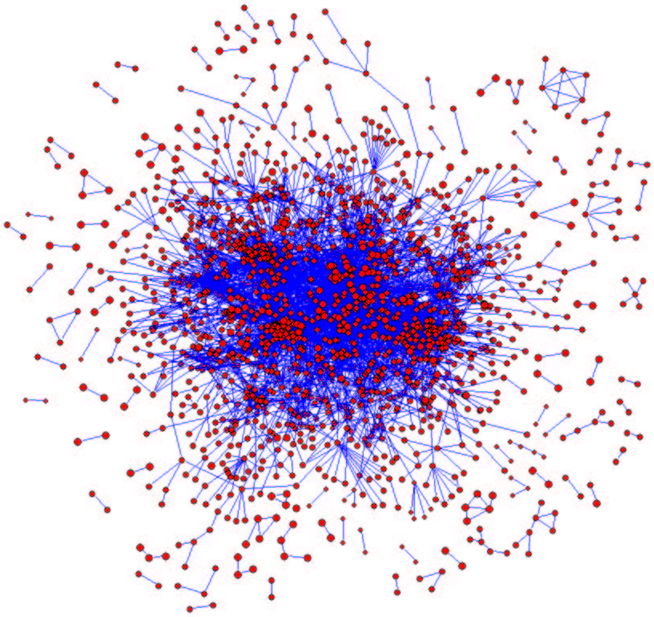

Figure 1: The PPI network constructed on $11,000$ yeast interactions [158] involving $2,401$ proteins.

One of the goals of systems biology is to explain relationships between structure, function, and regulation of molecular networks by combining theoretical and experimental approaches. To contribute to this goal, we focus our attention on analyzing structural properties of PPI networks and building predictive models for hypothesis generation. We give here an introduction to this multidisciplinary area of research. We first describe graph theoretic and biological terminology used in the literature and throughout this article. Then we survey three large research areas necessary for understanding the issues arising in studying PPI networks. We first give a survey of mathematical models of large networks and the most important properties of these models. Then we describe issues more closely related to PPI networks: PPI identification methods, publicly available PPI data sets, some of the biological structures embedded in the PPI networks and methods used for their detection, and the currently known mathematical properties of the currently available PPI networks. Then we give a brief survey of recent graph theoretic algorithms which have successfully been used in biological applications and which may be used to identify some of the biological structures in PPI networks. In the end, we point out interesting open problems which, if solved, may contribute to our understanding of cellular processes.

1.1 Graph Theoretic Terminology

We present here basic graph theoretic terminology used in this article, in agreement with West's textbook on graph theory [164].

A graph is a collection of points and lines connecting a subset of them; the points are called *nodes* or *vertices*, and the lines are called *edges*. A graph is usually denoted by G, or by $G(V, E)$, where V is the set of nodes and $E \subseteq V \times V$ is the set of edges of G. We often use n to represent $|V|$, and m to represent $|E|$. We also use $V(G)$ to represent the set of nodes of a graph G, and $E(G)$ to represent the set of edges of a graph G. A graph is *undirected* if its edges (node pairs) are undirected, and otherwise it is *directed*. Nodes joined by an edge are said to be *adjacent*. A *neighbor* of a node v is a node adjacent to v. We denote by $N(v)$ the set of neighbors of node v (called the *neighborhood* of v), and by $N[v]$ the *closed neighborhood* of v, which is defined as $N[v] = N(v) \cup \{v\}$. The *degree* of a node is the number of edges incident with the node. In directed graphs, an *in-degree* of a node is the number of edges ending at the node, and the *out-degree* is the number of edges originating at the node. A graph is *complete* if it has an edge between every pair of nodes. Such a graph is also called a *clique*. A complete graph on n nodes is commonly denoted by K_n. A graph G is *bipartite* if its vertex set can be partitioned into two sets, say A and B, such that every edge of G has one node in A and the other in B. A *path* in a graph is a sequence of nodes and edges such that a node belongs to the edges before and after it and no nodes are repeated; a path with k nodes is commonly denoted by P_k. The path *length* is the number of edges in the path. The shortest path length between nodes

u and v is commonly denoted by $d(u, v)$. The *diameter* of a graph is the maximum of $d(u, v)$ over all nodes u and v; if a graph is disconnected, we assume that its diameter is equal to the maximum of the diameters of its connected components. A *subgraph* of G is a graph whose nodes and edges all belong to G. An *induced subgraph* H of G, denoted by $H \lhd G$, is a subgraph of G on $V(H)$ nodes, such that $E(H)$ consists of all edges of G that connect nodes of $V(H)$. The *minimum edge cut* of a graph G is the set of edges S such that $|S|$ is of minimum size over all sets of edges that disconnect the graph upon removal. The minimum number of edges whose deletion disconnects G is called *edge connectivity*. A graph is k-edge-connected if its edge connectivity is $\geq k$. A graph is *weighted* if there is a weight function associated with its edges, or nodes. In this sense, the minimum weight edge cut of a weighted graph can be defined.

1.2 Biological Terminology

We give here definitions of basic biological terms used in this article. We assume that the definitions of DNA, RNA, protein, genome, and proteome are commonly known and do not include them here.

Proteins are important components of a cell. They are able of transferring signals, controlling the function of enzymes, regulating production and activities in the cell etc. To do this, they interact with other proteins, DNA, and other molecules. Some of the PPIs are permanent, while others happen only during certain cellular processes. Groups of proteins that together perform a certain cellular task are called *protein complexes*. There is evidence that protein complexes correspond to complete or “nearly complete” subgraphs of PPI networks (see section 3.3.1 and [132]). A *domain* is a part of a protein (and the corresponding segment of gene encoding the protein) that has its own function. The combination of domains in a protein determines its overall function. Examples of protein function include cell growth and maintenance, signal transduction, transcription, translation, metabolism, etc. [44]. Many domains mediate protein interactions with other biomolecules. A protein may have several different domains and the same domain may be found in different proteins.

A *molecular pathway* is a chain of cascading molecular reactions involved in cellular processes. Thus, they are naturally directed. Shortest paths in PPI networks have been used to model pathways (see section 3.3.2 and [132]).

Homology is a relationship between two biological features (here we consider genes, or proteins) which have a common ancestor. The two subclasses of homology are *orthology* and *paralogy*. Two genes are *orthologous* if they have evolved from a common ancestor by speciation; they often have the same function, taken over from the precursor gene in the species of origin. Orthologous gene products are believed to be responsible for essential cellular activities. In contrast, *paralogous* proteins have evolved by gene duplication; they either diverge functionally, or all but one of the versions is lost.

2 Large Network Models

A wide variety of systems can be described by complex networks. Such systems include: the cell, where we model the chemicals by nodes and their interactions by edges; the Internet, which is a complex network of routers and computers linked by various physical or wireless links; the World Wide Web, which is a virtual network of Web pages connected by hyper-links; and the food chain webs, the networks by which human diseases spread, the human collaboration networks etc. The emergence of the Internet, the World Wide Web, and the cellular function data made a big impact on modeling of large networks, which became a huge area of research on its own. Several articles give good surveys of large network models [122] [4] [121] [152]. We start with an overview of these survey articles and end the section with a presentation of more recent results.

Since the 1950s large networks with no apparent design have been modeled by random graphs, which represent the simplest model of a complex network. They were first studied by Erdos and Renyi [56], [57] [58], and later became a huge research area, a good survey of which was done by Bollobas [31]. They are based on the principle that the probability that there is an edge between any pair of nodes (denoted by p) is the same, distributed uniformly at random; thus, a random graph on n nodes has approximately $\frac{n(n-1)}{2}p$ edges, distributed uniformly at random. We describe random graphs in the next section.

There has been a growing interest in studying complex networks. As a result, it was shown that the topology of real-world networks differs from the topology of random graphs in several fundamental ways. Despite their large sizes, most real-world networks have relatively short paths between any two nodes. This property is often referred to as the *small-world* property. We will later see that under certain conditions random graphs satisfy this property. *Clustering* or *network transitivity* is the next characteristic of large networks: a network is said to show clustering if the probability of a pair of vertices being adjacent is higher when the two vertices have a common neighbor. Watts and Strogatz defined a *clustering coefficient C* as the average probability that two neighbors of a given vertex are adjacent [163]. More formally, if a node v in the network has d_v neighbors, the ratio between the number of edges E_v between the neighbors of v, and the largest possible number of edges between them, $\frac{d_v(d_v-1)}{2}$, is called the clustering coefficient of node v, and is denoted by C_v, where $C_v = \frac{2E_v}{d_v(d_v-1)}$. The clustering coefficient C of the whole network is the average of C_vs for all vertices v in the network. Real world complex networks exhibit a large degree of clustering, i.e., their clustering coefficient is large. However, this is not true for random graphs, since the probability that two of the neighbors of a vertex in a random graph are connected is equal to the probability that two randomly selected nodes are connected, and thus, $C = p$ for random graphs. The *degree distribution* is the next characteristic of large networks. If we denote by $P(k)$ the probability that a randomly selected vertex of a network has degree k, we can see that, since in a random graph edges are placed at random, the majority of nodes have the same degree which is close to the average degree of the graph. Thus, the degree distribution of a random graph is a Poisson distribution with a peak at the average degree of the graph. However, most real world large networks have a non-Poisson degree distribution. For example, a large number of these networks has the degree distribution with a power-law tail, $P(k) \approx k^{-\gamma}$. Such networks are called *scale free* [18].

These observations led to intensive research of network models over the past few years. The random graph model is still an active research area. The small-world model was motivated by clustering and it interpolates between the highly clustered regular ring lattices (defined below) and random graphs. The scale-free model was motivated by the discovery of the power-law degree distribution. We now review some of the main properties of each of these models.

2.1 Random Graphs

As mentioned above, one of the earliest theoretical models of a network was introduced and studied by Erdos and Renyi [56], [57] [58]. The model is called a *random graph* and it consists of n nodes joined by edges that have been chosen and placed between pairs of nodes uniformly at random. Erdos and Renyi gave several versions of their model, out of which the most commonly studied is the one denoted by $G_{n,p}$, where each possible edge in the graph on n nodes is present with probability p and absent with probability $1 - p$. The properties of $G_{n,p}$ are often expressed in terms of the average degree z of a vertex. The average number of edges in the graph $G_{n,p}$ is $\frac{n(n-1)}{2}p$, each edge contains two vertices, and thus the average degree of a vertex is $z = \frac{n(n-1)p}{n} = (n-1)p$, which is approximately equal to np for large n.

These types of graphs have many properties that can be calculated exactly in the limit of large n, which make them appealing as a model of a network. Thus, the following terminology is commonly used in the literature on random graphs: it is said that *almost all* random graphs (or *almost every* random graph) on n nodes have a property X, if the probability $Pr(X)$ that a graph has the property X satisfies $\lim_{n\to\infty} Pr(X) = 1$; similarly, a graph on n nodes *almost always,* or *almost surely,* satisfies a property X, if $\lim_{n\to\infty} Pr(X) = 1$. Examples of properties that can be calculated exactly in the limit of large n include the following. Erdos and

Renyi studied how the expected topology of a random graph changes as a function of the number of edges m. When m is small the graph is likely to be fragmented into many small connected components having vertex sets of size at most $O(\log n)$. As m increases the components grow at first by linking to isolated nodes, and later by fusing with other components. A transition happens at $m = \frac{n}{2}$, when many clusters cross-link spontaneously to form a unique largest component called the *giant component*, whose vertex set size is much larger than the vertex set sizes of any other components; the giant component contains $O(n)$ nodes, while the second largest component contains $O(\log n)$ nodes. Furthermore, the shortest path length between pairs of nodes in the giant component grows like $\log n$ (more details given later), and thus these graphs are small worlds. This result is typical for random-graph theory whose main goal usually is to determine at what probability p a certain graph property is most likely to appear. Erdos and Renyi's greatest discovery was that many important properties, such as the emergence of the giant component, appear quite suddenly, i.e., at a given probability either almost all graphs have some property, or almost no graphs have it.

The probability $P(k)$ of a given node in a random graph on n vertices having degree k is given by the binomial distribution:

$$P(k) = \left(\begin{array}{c} n-1 \\ k \end{array} \right) p^k (1-p)^{n-1-k},$$

which in the limit where $n \gg kz$ becomes the Poisson distribution $P(k) = \frac{z^k e^{-z}}{k!}$. Both of these distributions are strongly peaked around the mean z and have a tail that decays rapidly as $1/k!$. Minimum and maximum degrees of random graphs are determined and finite for a large range of p. For instance, if $p \approx n^{-1-1/k}$, almost no random graph has nodes with degree higher than k. On the other hand, if $p = \frac{\ln n + k \ln(\ln n) + c}{n}$, almost every random graph has a minimum degree of at least k. If $pn/\ln n \to \infty$, the maximum degree of almost all random graphs has the same order of magnitude as the average degree. Thus, a typical random graph has rather homogeneous degrees.

As mentioned above, random graphs tend to have small diameters. The range of values of the diameters of random graphs on n nodes and probability p is very narrow, usually concentrated around $\frac{\ln n}{\ln np} = \frac{\ln n}{\ln z}$ [42]. Other important results are that for $z = np < 1$, a typical random graph is composed of isolated trees and its diameter equals the diameter of a tree; if $z > 1$, the giant component emerges, and the diameter of the graph is equal to the diameter of the giant component if $z > 3.5$, and is proportional to $\frac{\ln n}{\ln k}$; if $z \geq \ln n$, almost every random graph is totally connected and the diameters of these graphs on n nodes and with the same z are concentrated on a few values around $\frac{\ln n}{\ln z}$. The average path length also scales with the number of nodes as $\frac{\ln n}{\ln z}$ and this is a reasonable first estimate for average path lengths of many real networks [121].

Random graphs have been extensively studied and a good survey of the area can be found in Bollobas's book [31]. They have served as idealized models of gene networks [86], ecosystems [104], and the spread of infectious diseases [85] and computer viruses [88]. We have seen above that even though some of their properties reasonably approximate the corresponding properties of real-world networks, they still differ from real-world networks in some fundamental ways. The first difference is in the degree distributions [6] [18]. As mentioned above, real networks appear to have power-law degree distributions [6] [60] [67] [5] [18], i.e., a small but not negligible faction of their vertices has a very large degree. These degree distributions differ from the rapidly decaying Poisson degree distribution, and they have profound effects on the behavior of the network. Examples of probability distributions of real-world networks are presented in Figure 2. The second difference between random graphs and real-world networks is the fact that real-world networks have strong clustering, while the Erdos and Renyi model does not [163] [162]. As mentioned above, in Erdos-Renyi random graphs the probabilities of pairs of vertices being adjacent are by definition independent, so the probability of two vertices being adjacent is the same regardless of whether they have a common neighbor, i.e., the clustering coefficient for a random graph is $C = p$. Table 1 is taken from [121] as an illustration of comparing clustering coefficients of real-world and random networks. It shows that random graphs do not provide an adequate model for real-world networks with respect to the network clustering property. Thus, we now turn to reviewing different network models which fit the real-world networks better.

network	n	z	C measured	C for random graph
Internet (autonomous systems) [128]	6,374	3.8	0.24	0.00060
World Wide Web (sites) [2]	153,127	35.2	0.11	0.00023
power grid[163]	4,941	2.7	0.080	0.00054
biology collaborations [115]	1,520,251	15.5	0.081	0.000010
mathematics collaborations [116]	253,339	3.9	0.15	0.000015
film actor collaborations [124]	449,913	113.4	0.20	0.00025
company directors [124]	7,673	14.4	0.59	0.0019
word co-occurrence [75]	460,902	70.1	0.44	0.00015
neural network [163]	282	14.0	0.28	0.049
metabolic network [61]	315	28.3	0.59	0.090
food web [113]	134	8.7	0.22	0.065

Table 1: For a number of different networks, n is the number of vertices, z is the mean degree, C is the clustering coefficient. Taken from [121].

2.2 Generalized Random Graphs

One approach to constructing a model with the degree distribution capturing the scale-free character of real networks is to allow a power-law degree distribution in a graph while leaving all other aspects as in the

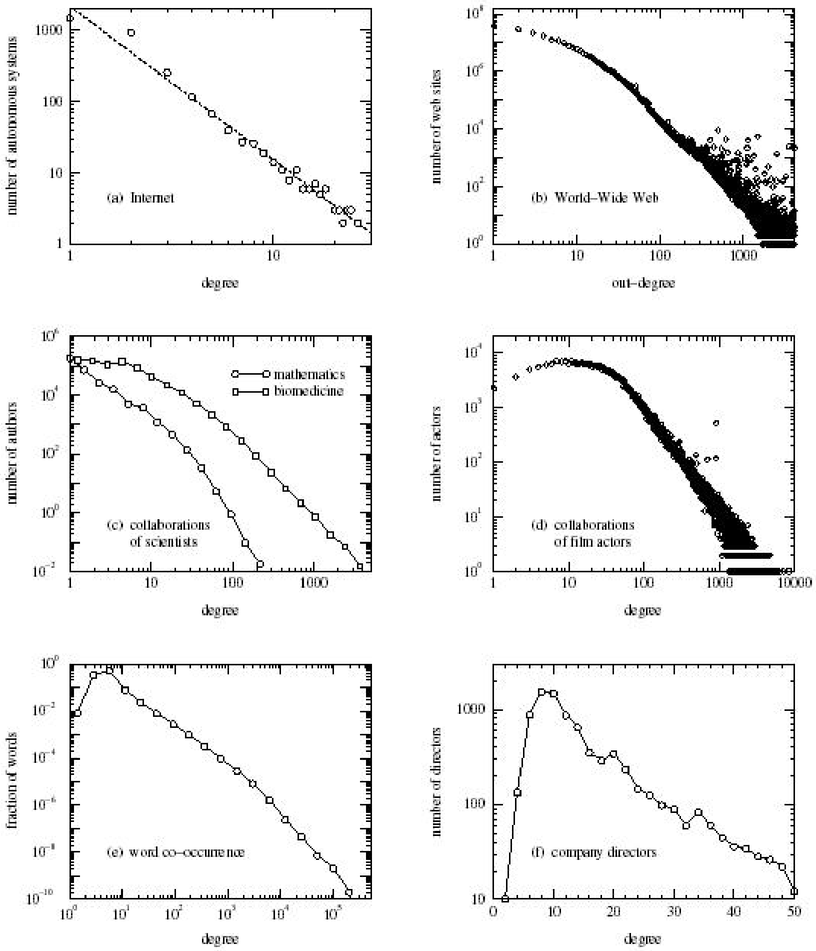

Figure 2: Degree distributions for different networks. (a) Physical connections between autonomous systems on the Internet in 1997 [60], (b) a 200 million page subset of the World Wide Web in 1999 [37], (c) collaborations between biomedical scientists and between mathematicians [115] [116], (d) collaborations of film actors [11], (e) co-occurrence of words in English [75], (f) board membership of directors of Fortune 1000 companies for year 1999 [124]. Taken from [121].

random graph model. That is, the edges are randomly chosen with the constraint that the degree distribution is restricted to a power law. These scale free random networks have been studied by systematically varying γ in the degree distribution $P(k) \approx k^{-\gamma}$ and observing if there is a threshold value of γ at which the properties of networks suddenly change.

Generating a random graph with a non-Poisson degree distribution is relatively simple and has been discussed a number of times in the literature. It appears that it has first been described by Bender and Canfield [28]. Given a degree sequence (i.e., distribution), one can generate a random graph by assigning to a vertex i a degree k_i from the given degree sequence, and then choosing pairs of vertices uniformly at random to make edges so that the assigned degrees remain preserved. When all degrees have been used up to make edges, the resulting graph is a random member of the set of graphs with the desired degree distribution. Of course, the sum of degrees has to be even to successfully complete the above algorithm. Note that this method does not allow the clustering coefficient to be specified, which is one of the crucial properties of these graphs that makes it possible to solve exactly for many of their properties in the limit of large n. For example, if we would like to find the mean number of second neighbors of a randomly chosen vertex in a graph with clustering, we have to account for the fact that many of the second neighbors of a vertex are also its first neighbors as well. However, in a random graph without clustering, the probability that a second neighbor of a vertex is also its first neighbor behaves as $\frac{1}{n}$ regardless of the degree distribution, and thus can be ignored in the limit of large n [121].

Recently, Luczak proved that almost all random graphs with a fixed degree distribution and no nodes of degree smaller than 2 have a unique giant component [95]. Molloy and Read [111] [112] derived a simple condition for the birth of the giant component, as well as an implicit formula for its size. More specifically, for $n \gg 1$ and $P(k) = \frac{d_k}{n}$ they defined $Q = \sum_{k=1}^{\infty} P(k)k(k-2)$ and showed that if $Q < 0$ the graph almost always consists of many small components, the average component size almost always diverges as $Q \to 0^-$, and a giant component almost surely emerges for $Q > 0$ under the condition that the maximum degree is less than $n^{\frac{1}{4}}$. Aiello, Chung, and Lu [3] applied these results to a random graph model for scale-free networks. They showed that for a power-law $P(k)$, the condition on Q implies that a giant component exists if and only if $\gamma < 3.47875... = \gamma_0$. When $\gamma > \gamma_0$, the random graph is disconnected and made of independent finite clusters, while when $\gamma < \gamma_0$ there is almost surely a unique infinite cluster. Aiello, Chung, and Lu studied the interval $0 < \gamma < \gamma_0$, and showed that for $2 \leq \gamma < \gamma_0$, the second largest component almost surely has a size of the order of $\ln n$. On the other hand, for $1 < \gamma < 2$, every node with degree greater than $\ln n$ almost surely belongs to the infinite cluster and the size of second largest component does not increase as the size of the graph goes to infinity; thus, the fraction of nodes in the infinite cluster approaches 1 as the system size increases meaning that the graph becomes totally connected in the limit of infinite system size. Finally, for $0 < \gamma < 1$, the graph is almost surely connected.

Newman, Strogatz, and Watts [124] developed a new approach to random graphs with a given degree distribution using a generating function formalism [165]. They have shown how, using the mathematics of generating functions, one can calculate exactly many of the statistical properties of these graphs in the limit of large n. They have given explicit formulas for the emergence of the giant component, the size of the giant component, the average distribution of the sizes of the other components, the average numbers of vertices at a certain distance from a given vertex, the clustering coefficient, the typical distance between a pair of vertices in a graph, etc. They started by defining the generating function $G_0(x) = \sum_{k=0}^{\infty} P(k)x^k$ for the probability distribution of vertex degrees k, where the distribution $P(k)$ is assumed to be normalized so that $G_0(1) = 1$. They derived the condition for the emergence of the giant component $\sum_k k(k-2)P(k) = 0$ identical to the one derived by Molloy and Read [111] (a positive sum leads to the appearance of a giant cluster), the size of the giant component, $S = 1 - G_0(u)$, where u is the smallest non-negative real solution of the equation $u = G_1(u)$, also identical to Molloy and Reed's [112], etc. They applied their theory to the modeling of collaboration graphs, which are bipartite, and the World Wide Web, which is directed, and showed that the theory gives good order of magnitude estimates of the properties of known collaboration graphs of business people, scientists, and movie actors, although there are measurable differences between theory and data that point to the presence of interesting effects, such as for example, sociological effects in collaboration networks.

2.3 Small-world Networks

As mentioned above, graphs that occur in many biological, social, and artificial systems often have a *small world* topology, i.e., a small-world character with unusually large clustering coefficients independent of network size. Watts and Strogatz proposed this one-parameter model of networks in order to interpolate between an ordered finite-dimensional lattice and a random graph [163]. They start from a ring lattice with n vertices and m edges in which every node is adjacent to "its first k neighbors" on the ring (an illustration is presented in Figure 3), and "re-wire" each edge at random with probability p, not allowing for self-loops and multiple edges. This process introduces $\frac{pnk}{2}$ "long-range" edges. Thus, the graph can be "tuned" between regularity ($p = 0$) and disorder ($p = 1$).

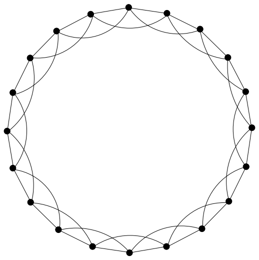

Figure 3: A regular ring lattice for $k = 2$.

Watts and Strogatz quantified the structural properties of these graphs by their *characteristic path length* $L(p)$ (the shortest path length between two vertices averaged over all pairs of vertices) and the above mentioned clustering coefficient $C(p)$ (remember that if $C_v = \frac{|E(N(v))|}{\frac{1}{2}|N(v)|(|N(v)|-1)}$, then C is the average of C_v

over all vertices v) as functions of re-wiring probability p. They established that the regular lattice at $p = 0$ is a highly clustered "large world" in which L grows linearly with n, since $L(0) \approx \frac{n}{2k} \gg 1$ and $C(0) \approx \frac{3}{2}$. On the other hand, as $p \rightarrow 1$ the model converges to a random graph, which is a poorly clustered "small world" where L grows logarithmically with n, since $L(1) \approx \frac{\ln n}{\ln k}$, and $C(1) \approx \frac{k}{n}$. Note that these limiting cases do not imply that large C is always associated with large L, and small C with small L. On the contrary, they found that even the slightest amount of re-wiring transforms the network into a small world with short paths between any two nodes, like in the giant component of a random graph, but at the same time such a network is much more clustered than a random graph. This is in excellent agreement with the characteristics of real-world networks. They have shown that the collaboration graph of actors in feature films, the neural network of the nematode worm *C. elegans*, and the electrical power grid of the western United States all have a small world topology, and they conjectured that this model is generic for many large, sparse networks found in nature. Since then many empirical examples of small-world networks have been documented [18] [82] [159] [11] [149] [146].

The pioneering paper of Watts and Strogatz started a large area of research. Walsh [161] showed that graphs associated with many different search problems have a small-world topology, and that the cost of solving them can have a heavy-tailed distribution. This is due to the fact that local decisions in a small-world topology quickly propagate globally. He proposes randomization and restarts to eliminate these heavy tails. Kleinberg [90] proved that the problem of how to find a short chain of acquaintances linking oneself to a random person using only local information (this problem was originally posed by Milgram's original sociological experiment [109]) is easily solvable only for certain kinds of small worlds.

There has been a lot of research in small-world networks outside computer science as well. Epidemiologists have wondered how local clustering and global contacts together influence the spread of infectious disease, trying to make vaccination strategies and understand evolution of virulence [160] [17] [87] [33]. Neurobiologists have asked about possible evolutionary significance of small-world neural topology. They have argued that small-world topology combines fast signal processing with coherent oscillations [94] and thus was selected by adaptation to rich sensory environments and motor demands [24].

The most active research of small-world networks happened in statistical physics. A good review can be found in Newman's review article [119]. A variant of the Watts-Strogatz model was proposed by Newman and Watts [117] [118] in which no edges are removed from the regular lattice and new edges are added between randomly chosen pairs of nodes. This model is easier to analyze, since it does not lead to the formation of isolated components, which could happen in the original model. Newman, Moore, and Watts derived the formula for a characteristic path length in these networks, $L(p) = \frac{n}{k} f(nkp)$, where $f(x) \approx \frac{1}{2\sqrt{x^2+2x}} \tanh^{-1} \frac{x}{\sqrt{x^2+2x}}$ [123]. This solution is asymptotically exact in the limits of large n and when either $nkp \to \infty$, or $nkp \to 0$ (large or small number of shortcuts). Barbour and Reinert [22] improved this result by finding a rigorous distributional approximation for $L(p)$ together with a bound on the error.

Small-world networks have a relatively high clustering coefficient. In a regular lattice ($p = 0$), the clustering coefficient does not depend on the lattice size, but only on its topology. It remains close to $C(0)$ up to relatively large values of p as the network gets randomized. Using a slightly different, but equivalent definition of C, Barrat and Weigt [23] have derived a formula for $C(p)$. According to their definition, $C'(p)$ is the fraction between the mean number of edges between the neighbors of a node and the mean number of possible edges between those neighbors. Starting with a regular lattice with a clustering coefficient $C(0)$, and observing that for $p > 0$ two neighbors of a node v that were connected at $p = 0$ are still neighbors of v and connected by an edge with probability $(1-p)^3$, since there are three edges that need to remain intact, we conclude that $C'(p) \approx C(0)(1-p)^3$. Barrat and Weigt [23] verified that the deviation of $C(p)$ from this expression is small and goes to zero as $n \to \infty$. The corresponding expression for the Newman-Watts model is $C'(p) = \frac{3k(k-1)}{2k(2k-1)+8pk^2+4p^2k^2}$ [120].

The degree distribution of small-world networks is similar to that of a random graph. In the Watts-Strogatz model for $p = 0$, each node has the same degree k. A non-zero p introduces disorder in the network and widens the degree distribution while still maintaining the average degree equal to k. Since only one end of an edge gets re-wired, $\frac{pnk}{2}$ edges in total, each node has degree at least $\frac{k}{2}$ after re-wiring. Thus, for $k > 2$ there are no isolated nodes. For $p > 0$, the degree k_v of a vertex v can be expressed as $k_v = \frac{k}{2} + c_v$ [23], where c_v is divided into two parts, $c_v = c_v^1 + c_v^2$, so that $c_v^1 \leq \frac{k}{2}$ edges have been left in place with probability $1-p$, and c_v^2 edges have been re-wired towards v, each with probability $\frac{1}{n}$. For large n the probability distributions for c_v^1 and c_v^2 are $P_1(c_v^1) = \binom{\frac{k}{2}}{c_v^1}(1-p^{c_v^1})p^{\frac{k}{2}-c_v^1}$ and $P_2(c_v^2) = \binom{\frac{pnk}{2}}{c_v^2}(\frac{1}{n})^{c_v^2}(1-\frac{1}{n})^{\frac{pnk}{2}-c_v^2} \approx \frac{(pk/2)^{c_v^2}}{c_v^2!}e^{-pk/2}$. Combining these two factors, the degree distribution is $P_p(c) = \sum_{n=0}^{min(c-k/2,k/2)} \binom{k/2}{n}(1-p)^n p^{k/2-n} \times \frac{(pk/2)^{c-k/2-n}}{(c-k/2-n)!}e^{-pk/2}$, for $c \geq \frac{k}{2}$. As p grows, the distribution becomes broader, but it stays strongly peaked at the average degree with an exponentially decaying tail.

2.4 Scale-free Networks

Many real networks have a property that some nodes are more highly connected than the others. For example, the degree distributions of the Internet backbone [60], metabolic reaction networks [82], the telephone call graph [1], and the World Wide Web [37] decay as a power law $P(k) \approx k^{-\gamma}$, with the exponent

$\gamma \approx 2.1 - 2.4$ for all of these cases. This form of heavy-tailed distribution would imply an infinite variance, but in reality there are a few nodes with many links, such as, for example, search engines for the World Wide Web.

The earliest work on the theory of scale-free networks was due to Simon [144] [34] in 1955, and it was recently independently discovered by Barabasi, Albert, and Jeong [18] [19]. They showed that a heavy-tailed degree distribution emerges automatically from a stochastic growth model in which new nodes are added continuously and they preferentially attach to existing nodes with probability proportional to the degree of the target node. That is, high-degree nodes become of even higher degree with time and the resulting degree distribution is $P(k) \approx k^{-3}$. They also showed that if either the growth, or the preferential attachment are eliminated, the resulting network does not exhibit scale-free properties. That is, both the growth and preferential attachment are needed simultaneously to produce the power-law distribution observed in real networks.

The average path length in the Barabasi-Albert network is smaller than in a random graph, indicating that a heterogeneous scale-free topology is more efficient in bringing nodes close together than the homogeneous random graph topology. Recent analytical results show that the average path length, ℓ, satisfies $\ell \approx \frac{\ln n}{\ln \ln n}$ [32]. Another interesting phenomenon is that while in random graph models with arbitrary degree distribution the node degrees are uncorrelated [3] [124], non-trivial correlations develop spontaneously between the degrees of connected nodes in the Barabasi-Albert model [91]. There has been no analytical prediction for the clustering coefficient of the Barabasi-Albert model. It has been observed that the clustering coefficient of a scale-free network is about five times higher than that of a random graph, and that this factor slowly increases with the number of nodes [4]. However, the clustering coefficient of the Barabasi-Albert model decreases with the network size approximately as $C \approx n^{-0.75}$, which is a slower decay than the $C = \frac{<k>}{N}$ for random graphs, where $< k >$ denotes the average degree, but is still different from the small-world models in which C is independent of n.

The Barabasi-Albert model is a minimal model that captures the mechanisms responsible for the power-law degree distributions observed in real networks. The discrepancies between this model and real networks, such as the fixed exponent of the predicted power-law distribution for the model, while real networks have measured exponents varying between 1 and 3, led to a lot of interest in addressing network evolution questions. The theory of evolving networks emerged offering insights into network evolution and topology. More sophisticated models including the effects of adding or re-wiring edges, allowing nodes to age so that they can no longer accept new edges, or varying the form of preferential attachment have been developed [5] [48]

[92]. In addition to scale-free degree distributions, these generalized models also predict exponential and truncated power-law degree distribution in some parameter regimes. Albert, Jeong, and Barabasi [7] suggest that scale-free networks are resistant to random failures due to a few high-degree "hubs" dominating their topology: any node that fails probably has a small degree, and thus does not severely affect the rest of the network. However, such networks are vulnerable to deliberate attacks on the hubs. These intuitive ideas have been confirmed numerically [37] [7] and analytically [43] [39] by examining how the average path length and size of the giant component depend on the number and degree of the removed nodes. Implications have been made for the resilience of the Internet [34], the design of therapeutic drugs [82], and the evolution of metabolic networks [82] [159].

To generate networks with scale-free topologies in a deterministic, rather than stochastic way, Barabasi, Ravasz, and Vicsek [21] have introduced a simple model, which they solved exactly showing that the tail of the degree distribution of the model follows a power law. The first steps of the construction are presented in Figure 4. The construction can be viewed as follows. The starting point is a P_3. In the next iteration, add two more copies of the P_3 and connect the mid-point of the initial P_3 with the outer nodes of the two new P_3s. In the next step, make two copies of the 9-node module constructed in previous step, and connect "end" nodes of the two new copies to the "middle" node of the old module (as presented in Figure 4). This process can continue indefinitely. They showed that the degree probability distribution of such a graph behaves as $P(k) \approx k^{\frac{\ln 3}{\ln 2}}$. An additional property that these networks have is the hierarchical combination of smaller modules into larger ones. Thus, they called these networks "hierarchical".

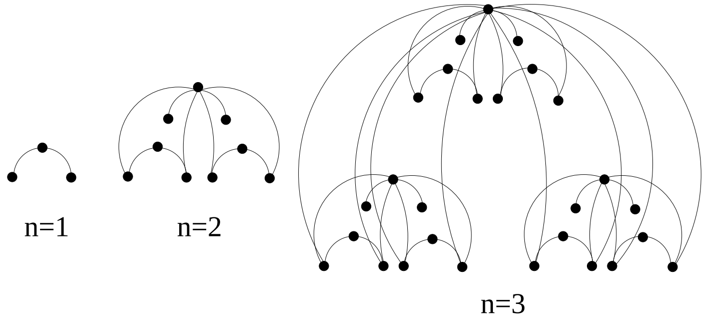

Figure 4: Scheme of the growth of a scale-free deterministic hierarchical graph. Adapted from [21].

Recently, Dorogovtsev, Goltsev, and Mendes [47] have proposed another deterministic graph construction to model evolving scale-free networks, which they called "pseudofractal". The scheme of the growth of the scale-free pseudofractal graph is presented if Figure 5. They showed that the degree distribution of their graph can be characterized by a power law with exponent $\gamma = 1 + \ln 3/\ln 2 \approx 2.585$, which is close to the distribution of real growing scale-free networks. They found, both exactly and numerically, all main characteristics of their graph, such as the shortest path length distribution following a Gaussian of width $\approx \sqrt{\ln n}$ centered at $\bar{l} \approx \ln n$ (for $\ln n \gg 1$), clustering coefficient of a degree k vertex following $C(k) = 2/k$, and the eigenvalue spectrum of the adjacency matrix of the graph following a power-law with the exponent $2 + \gamma$.

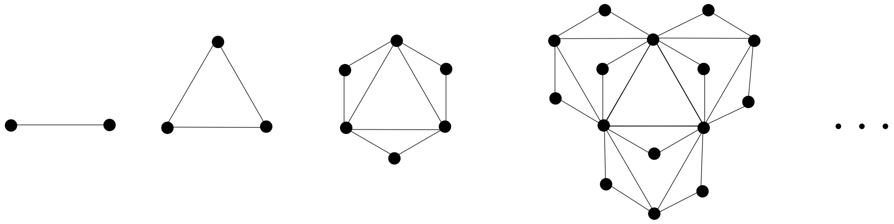

Figure 5: Scheme of the growth of the scale-free pseudofractal graph. The graph at time step $t + 1$ can be constructed by "connecting together" three graphs from step t. Adapted from [47].

3 Protein Interaction Networks

With vast amounts of DNA sequences becoming available in recent years, there is a growing interest in correlating the genome with the proteome to explain biological function and to develop new effective protein targeting drugs. One of the key questions in proteomics today is with which proteins does a certain protein interact. The hope is to exploit this information for therapeutic purposes. Different methods have been used to identify protein interactions, including biochemical as well as computational approaches. A survey of biochemical methods used to identify proteins and PPIs can be found in Pandey and Man's article [127]. Most of them are lab intensive and of low accuracy. Despite this low confidence in the identified PPIs, maps of protein-protein interactions are being constructed and their analysis is attracting more and more attention. Many laboratories throughout the world are contributing to one of the ultimate goals of the biological sciences – the creation and understanding of a full molecular map of a cell. To contribute to these efforts, we focus our attention to studying currently available networks of PPIs.

Despite many shortcomings of representing a PPI network using the standard mathematical representation of a network, with proteins being represented by nodes and interactions being represented by edges, this has been the only mathematical model used so far to represent and analyze these networks. This model does not address the following important properties of PPI data sets. First, there is a large percentage of false-positive interactions in these data sets. For example, a common class of false-positive PPIs happens when in reality proteins indirectly interact, i.e., through mediation of one or more other molecules, but an experimental method detects this as a direct physical interaction. This may be a reason why very dense subnetworks are being detected inside PPI networks. False-negative interactions are also present in these networks resulting from non-perfect experimental techniques used to identify interactions. Other drawbacks of the model include the following PPI properties not being captured by this model: spatial and temporal information, information about experimental conditions, strength of the interactions, number of experiments confirming the interaction etc. The last point has been addressed by von Mering *et al.* [158] who classified PPIs into groups depending on the number of experiments that detected a specific PPI; they call this a *level of confidence* that a given PPI is a true interaction.

In this section we give an overview of recent PPI identification methods, currently available PPI data sets and their repositories, biological structures contained in the PPI networks (such as protein complexes and pathways), and computational methods used to identify them in PPI networks. Then we focus on surveying the literature on PPI network properties and structure. We point to open problems and future research directions in the area of PPI networks in section 5.

3.1 PPI Identification Methods

As mentioned above, the lists of genes and encoded proteins are becoming available for an increasing number of organisms. Databases such as Ensembl [72] and GenBank [29] (described in the next sub-section) contain publicly available DNA sequences for more than 105,000 organisms, including whole genomes of many organisms in all three domains of life, bacteria, archea, and eukaryota, as well as their protein data. In parallel to the increasing number of genomes becoming available, high-throughput protein-protein interaction detection methods have been introduced in the past couple of years producing a huge amount of interaction data. Such methods include yeast two-hybrid systems [156] [77] [76], protein complex purification methods using mass spectrometry [63] [71], correlated messenger RNA (m-RNA) expression profiles [73], genetic interactions [108], and *in silico* interaction predictions derived from gene fusion [54], gene neighborhood [46], and gene co-occurrences or phylogenetic profiles [74]. None of these PPI detection methods is perfect and the rates of false positives and false negatives vary from method to method. A brief summary describing these methods can be found in the article by von Mering *et al.* [158]. We outline here the main characteristics of each of these methods following [158].

Yeast two-hybrid assay is an *in vivo* technique involving fusing one protein to a DNA-binding domain and the other to a transcriptional activator domain. An interaction between them is detected by the formation of a functional transcription factor. This technique detects even transient and unstable interactions. However, it is not related to the physiological setting. *Mass spectrometry of purified complexes* involves tagging individual proteins which are used as hooks to biochemically purify whole protein complexes. The complexes are separated and their components identified by mass spectrometry. There exist two protocols, tandem affinity purification (TAP) [135] [63], and high-throughput mass-spectrometric protein complex identification (HMS-PCI) [97] [71]. This technique detects real complexes in their physiological settings and enables a consistency check by tagging several members of a complex at the same time. However its drawbacks are that it might miss some complexes that are not present under the given conditions, tagging can disturb complex formation, and loosely associated components can be washed off during purification. *Correlated m-RNA expression (synexpression)* involves measuring m-RNA levels under many different cellular conditions and grouping genes which show a similar transcriptional response to these conditions. The groups that encode physically interacting proteins were shown to frequently exhibit this behavior [64]. This is an indirect *in vivo* technique which has a much broader coverage of cellular conditions than other techniques. However, it is very sensitive to parameter choices and clustering methods used during the analysis, and thus is not very accurate for predicting direct physical interactions. *Genetic interactions* is an indirect *in vivo* technique which involves detecting interactions by observing the phenotypic results of gene mutations. An example of a genetic interaction is *synthetic lethality* which involves detecting pairs of genes that cause lethality when mutated at the same time. These genes are frequently functionally associated and thus their encoded proteins may physically interact. *In silico* predictions through genome analysis involve screening whole genomes for the following types of interaction evidence: (a) finding conserved operons in prokaryotic genomes which often encode interacting proteins [46]; (b) finding similar phylogenetic profiles, since interacting proteins often have similar phylogenetic profile, i.e., they are either both present, or both absent from a fully sequenced genome [74]; (c) finding proteins that are found fused into one polypeptide chain [54]; (d) finding structural and sequence motifs within the protein-protein interfaces of known interactions that allow the construction of general rules for protein interaction interfaces [83] [84]. *In silico* methods are fast and inexpensive techniques whose coverage expands with more and more organisms being fully sequenced. However, they require orthology between proteins and fail when orthology relationships are not clear.

3.2 Public Data Sets

Vast amounts of biological data that are constantly being generated around the world are being deposited in numerous databases. There are still no standards for accumulation of PPI data into databases. Thus, different PPI databases contain PPIs from different single experiments, high-throughput experiments, and literature sources. PPIs resulting from the most recent studies are usually only available on the journal web sites where the corresponding papers appeared. Here we briefly mention the main databases, including nucleotide sequence, protein sequence, and PPI databases. Nucleotide and protein sequence databases do not stuffer from the lack of standardization that is present in PPI databases. A recent comprehensive list of major molecular biology databases can be found in recent Baxevanis's article [26].

The largest nucleotide sequence databases are EMBL [1] [151], GenBank [2] [29], and DDBJ [3] [153]. They contain sequences from the literature as well as those submitted directly by individual laboratories. These databases store information in a general manner for all organisms. Organism specific databases exist for many organisms. For example, the complete genome of bakers yeast and related yeast strains can be found in Saccharomyces Genome Database (SGD) [4] [50]. FlyBase [5] [12] contains the complete genome of the fruit fly *Drosophila melanogaster*. It is one of the earliest model organism databases. Ensembl [6] [72] contains the draft human genome sequence along with its gene prediction and large scale annotation. It currently contains over 4,300 megabases and 29,000 predicted genes, as well as information about predicted genes and proteins, protein families, domains etc. Ensembl is not only free, but is also open source.

SwissProt [7] [16] and Protein Information Resource (PIR) [8] [105] are two major protein sequence databases. They are both manually curated and contain literature links. They exhibit a large degree of overlap, but still contain many sequences that can be found in only one of them. SwissProt maintains a high level of annotations for each protein including its function, domain structure, and post-translational modification information. It contains over 101,000 curated protein sequences. Computationally derived translations of EMBL nucleotide coding sequences that have not yet been integrated into the SwissProt resource can be found in Trembl [9]. The Non-Redundant Database (NRDB) [10] merges two sequences into a representative sequence if they exhibit a large degree of similarity. This is useful when a large scale computational analysis needs to be performed.

Some of the main databases containing protein interaction data are the following. The Munich Information Center for Protein Sequences (MIPS) [11] [108] provides high quality curated genome related information, such as protein-protein interactions, protein complexes, pathways etc., spanning over several

[1] http://www.ebi.ac.uk/embl/

[2] http://www.ncbi.nlm.nih.gov/Genbank/

[3] http://www.ddbj.nig.ac.jp/

[4] http://genome-www.stanford.edu/Saccharomyces/

[5] http://flybase.bio.indiana.edu/

[6] http://www.ensembl.org/

[7] http://www.ebi.ac.uk/swissprot/

[8] http://pir.georgetown.edu/

[9] http://www.ebi.ac.uk/trembl/

[10] http://www.ebi.ac.uk/ holm/nrdb90

[11] http://mips.gsf.de

organisms. Yeast Proteomics Database (YPD) [12] [45] is another curated database. It contains bakers yeast, *S. cerevisiae*, protein information, including their sequence and genetic information, related proteins, PPIs, complexes, literature links, etc. The Database of Interacting Proteins (DIP) [13] [168] is a curated database containing information about experimentally determined PPIs. It catalogues around $11,000$ unique interactions between $5,900$ proteins from over 80 organisms [168] including yeast and human. The Biomolecular Interaction Network Database (BIND) [14] [13] archives biomolecular interaction, complex, and pathway information. In this database, the biological objects interacting could be: a protein, RNA, DNA, molecular complex, small molecule, photon, or gene. This database includes Pajek [25] as a network visualization tool. It includes a network clustering tool as well, called the Molecular Complex Detection (MCODE) algorithm [15] (described in section 3.3), and a functional alignment search tool (FAST) [13] (details of FAST are not yet available in the literature) which displays the domain annotation for a group of functionally related proteins. The General Repository for Interaction Datasets (GRID) [15] [35] is a database of genetic and physical interactions which contains interactions from several genome and proteome wide studies, as well as the interactions from MIPS and BIND databases. It also provides a powerful network visualization tool called Osprey [36].

A recent study of the quality of yeast PPIs was done by von Mering *et al.* [158]. They have performed a systematic synthesis and evaluation of PPIs detected by major high-throughput PPI identification methods for yeast *S. Cerevisiae*, a model organism relevant to human biology [137]. They integrated $78,390$ interactions between $5,321$ yeast proteins, out of which only $2,455$ are supported by more than one method. This low overlap between the methods may be due to a high rate of false positives (rates of false positives and false negatives differ from method to method), or to difficulties in detecting certain types of interactions by specific methods. Also, certain research groups are interested in finding interactions between certain types of proteins which contributes to the lack of overlap between different PPI data sets. Von Mering *et al.* have found that each PPI identification technique produces a unique distribution of interactions with respect to functional categories of interacting proteins. They assessed the quality of interaction data and produced a list of $78,390$ yeast PPIs ordered by the level of confidence (high, medium, and low) with the highest confidence being assigned to interactions confirmed by multiple methods. Their list of PPIs currently represents the largest publicly available collection of PPIs for *S. Cerevisiae*, and also the largest PPI collection for any organism.

[12] http://www.incyte.com/sequence/proteome/databases/YPD.shtml

[13] http://dip.doe-mbi.ucla.edu/

[14] http://www.binddb.org/

[15] http://biodata.mshri.on.ca/grid/

3.3 Biological Structures Within PPI Networks and Their Extraction

We focus here only on protein complexes and pathways. We first briefly describe the most recent biochemical studies that have been used to identify these biological structures. These studies are expensive, time consuming, and of low accuracy. Computational detection of biological structures from PPI networks may supplement these approaches to reduce their time and cost, and increase their accuracy. The hope is that with the emergence of high confidence PPI networks, such as, for example, the one constructed on high-confidence PPIs from the study of von Mering *et al.* [158], computational approaches will become inexpensive and reliable tools for extraction of known and prediction of still unknown members of these structures. Despite a large body of literature involving purely theoretical aspects of networks, such as, for example finding clusters in graphs (see section 4), very few of such methods have been developed specifically for biological applications and applied to PPI networks. We outline the more successful ones below.

3.3.1 Protein Complexes

As mentioned earlier, cellular processes are usually carried out by groups of proteins acting together to perform a certain function. These groups of proteins are commonly called *protein complexes*. They are not necessarily of invariable composition, i.e., a complex may have several core proteins which are always present in the complex, as well as more dynamic, perhaps regulatory proteins, which are only present in a complex from time to time. Also, the same protein may participate in several different complexes during different cellular activities. One of the most challenging aspects of PPI data analysis is determining which of the myriad of interactions in a PPI network comprise true protein complexes [71] [52] [154].

Recently, mass spectrometry studies have been conducted to identify protein complexes in yeast *S. cerevisiae*. Ho *et al.* used HMS-PCI to extract complexes from *S. cerevisiae* proteome [71]. They reported approximately threefold higher success rate in detection of known complexes when compared to the large-scale two-hybrid studies by Uetz *et al.* [156] and Ito *et al.* [76]. Gavin *et al.* have performed an analysis of the *S. cerevisiae* proteome as a network of protein complexes [63]. They used the mass spectrometry approach to identify protein complexes which yielded about 70% probability of detecting the same protein in two different purifications. Amongst 1,739 yeast genes, including 1,143 human orthologues, they purified 589 protein assemblies, out of which 98 corresponded to protein complexes in the Yeast Protein Database (YPD), 134 were new complexes, and the remaining ones showed no detectable association with other proteins. This led to proposing a new cellular function for 344 proteins including 231 proteins with no previous functional annotation. They attempted investigating relationships between complexes in order to understand the integration and coordination of cellular functions by representing each complex as a node and having an edge between two nodes if the corresponding complexes share proteins. Then they color-coded complexes according to their cellular roles and noticed grouping of the same colored complexes, suggesting that sharing of components reflects functional relationships. No graph theoretic analysis of this protein complex network has been done so far. They also compared human and yeast complexes and found that orthologous proteins preferentially interact with complexes enriched with other orthologues, supporting the existence of "orthologous proteome" which may represent core functions for all eukaryotic cells.

There have been a couple of attempts to computationally extract protein complexes out of a PPI network. They involved measurements of connectedness (e.g., k-core concept [14]), Watts-Strogatz's vertex neighborhood "cliquishness" [163] (e.g., MCODE method [15]), or the reliance on reciprocal bait-hit interactions as a measure of complex involvement. Bader and Hogue [15] designed a simple algorithm that they termed the "Molecular Complex Detection" (MCODE) algorithm. The algorithm exploits Watts and Strogatz's notion of the clustering coefficient described in Section 2. They used the notion of a *k-core*, a graph of minimum degree k, and a notion of the "highest k-core of a graph", the most densely connected k-core of a graph, to weight a PPI network vertices in the following way. They define a core-clustering coefficient of a vertex v to be the density of the highest k-core of $N[v]$, where density is the number of edges of a graph divided by the maximum possible number of edges of the graph. The weight a vertex v is the product of the vertex core-clustering coefficient and the highest k-core level, k_{max}, of the $N(v)$. Then they seed a complex in the weighted graph with the highest weighted vertex and recursively move outward from the seed vertex including in the complex vertices with weights above a given threshold. They repeat this for the next highest weighted unexplored vertex. After this they do post-processing by discarding complexes that do not contain a k-core with $k \geq 2$. They also include the following two options: an option to "fluff" the complexes by adding to them their neighbors unexplored by the algorithm of weight bigger than the "fluff" parameter, and an option to "haircut" the complex by removing vertices of degree 1 from the complex. The resulting complexes are scored according to the product of the complex vertex set size and the complex density, and they are ranked according to the scoring function. Their algorithm also offers an option to specify a seed vertex. They evaluated their algorithm against Gavin's and MIPS complexes data sets. They tried all combinations of their parameters (true/false for haircut and fluff, and vertex weight percentage in 0.05 increments) to find the combination of parameters that yields the largest overlap with known protein complexes. They used 221 Gavin's complexes to evaluate MCODE and found that MCODE complexes matched only 88 out of the 221 Gavin's complexes. They obtained a similar discouraging result for MIPS, where MCODE predicted 166 complexes out of which only 52 matched MIPS complexes. Their complexes of high density were highly likely to match real complexes. This points out that a different approach of finding efficient graph clustering algorithms that would identify highly connected subgraphs should be taken to identify protein complexes in PPI networks. We explored this approach and used Hartuv and Shamir's Highly Connected Subgraph (HCS) algorithm [70] [69] (described in section 4) to identify protein complexes from yeast PPI network with 11,000 interactions amongst 2,401 proteins [132]. The algorithm detected a number of known protein complexes (an illustration is presented in Figure 6). Also, 27 out of 31 clusters identified in this way had high overlaps with protein complexes documented in MIPS database. The remaining 4 clusters that did not overlap MIPS contained a functionally homogeneous 6-protein cluster Rib 1-5 and a cluster Vps20, 25, 36, which are likely to correspond to protein complexes. In addition, the clusters identified in this way had a statistically significant functional homogeneity.

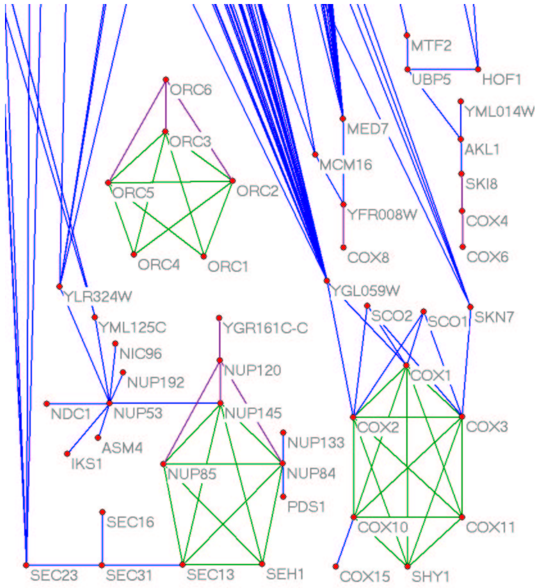

Figure 6: A subnetwork of a yeast PPI network [132] showing some of the identified complexes (green). Violet lines represent PPIs to proteins not identified as biological complex members due to stringent criteria about their connectivity in the algorithm, or due to absence of protein interactions that would connect them to the identified complex (from more details see [132]).

An approach similar to ours has been taken by Spirin, Zhao, and Mirny [145]. They explored several techniques for identification of highly connected subgraphs in a PPI network [145]. Their results are consistent with our analysis [132]. They used three different methods to identify protein complexes from the PPI network constructed on MIPS database PPIs. Their first method involves identifying complete subgraphs of the PPI graph. The second method they used is the Super-Paramagnetic Clustering algorithm developed by

Blatt, Wiseman, and Domany [30] to cluster objects in a non-metric space of an arbitrary dimension. As the third method, they developed their own novel Monte-Carlo optimization technique to identify highly connected subgraphs in a network (the details of this algorithm have not been published yet). They reported that most of the dense subnetworks that they identified in this way had consistent functional annotation revealing the function of the whole complex. Also, their dense subgraphs had a good agreement with the known protein complexes from MIPS, BIND, and the study of Ho *et al.* [71]. They claim to have also predicted several novel complexes and pathways, but they do not give any further details on these.

Bu *et al.* [38] used a similar approach to predict functions of uncharacterized proteins. They applied spectral graph theory methods, that have previously been used for analyzing the World Wide Web [65] [89], to the yeast PPI network constructed on high and medium confidence interactions from von Mering's paper [158]. They identified "quasi-cliques" and "quasi-bipartites" in the PPI network and noticed that proteins participating in quasi-cliques usually share common functions. They subsequently assigned functions to 76 uncharacterized proteins.

3.3.2 Molecular Pathways

Molecular *pathways* are chains of cascading molecular reactions involved in maintaining life. Different processes involve different pathways. Some examples include metabolic pathways, apoptosis pathways, or signaling pathways for cellular responses. An example of a signaling pathway transmitting information from the cell surface to the nucleus where it causes transcriptional changes, is presented in Figure 7. Disruption in a pathway function may cause severe diseases, such as cancer. Thus, understanding molecular pathways is an important step in understanding cellular processes and the effects of drugs on cellular processes. As a consequence, modeling and extraction of pathways from a network of protein interactions has become a very active research area. The Biopathways Consortium [16] was founded to catalyze the emergence and development of computational pathways biology. One of their main goals is to coordinate development and use of open technologies, standards, and resources for representing, handling, accessing, and analyzing pathways information. Numerous papers addressing these topics have been presented at the recent *4th and 5th Biopathways Consortium Meeting*. Many of them used classical graph algorithms in order to integrate genome-wide data on regulatory proteins and their targets with protein-protein interaction data in yeast [169], reconstruct microbial metabolic pathways [106], determine parts of structure and evolution of metabolic networks [96] etc.

[16] http://www.biopathways.org/

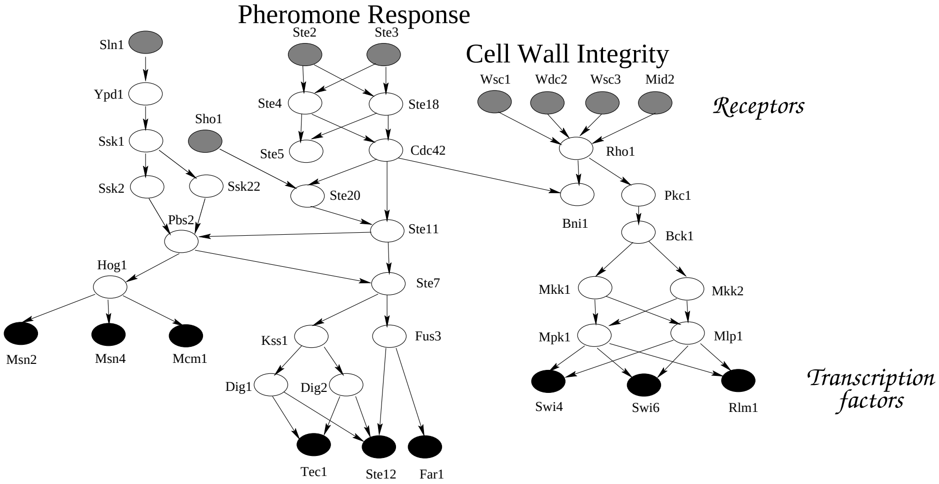

Figure 7: Examples of MAPK (mitogen-activated protein kinases) signal transduction pathways in yeast. Gray nodes represent membrane proteins, black nodes represent transcription factors, and white nodes represent intermediate proteins.

Recently, Steffen *et al.* constructed a model of *S. cerevisiae* signal transduction networks using simple graph theory [147]. They used yeast two-hybrid PPI data [156] [76] [139] to form a PPI network in which they deleted the most highly connected nodes, and then identified shortest paths of length at most eight between every membrane protein and every DNA-binding protein in the modified network. They deleted the most highly connected nodes in order to reduce the number of candidate signaling pathways from around 17 million to around 4.4 million. They compared these pathways with the ones obtained in the same way in three randomized PPI networks, and tuned the parameters (the number of clusters in which genes were grouped, the microarray expression datasets used in clustering, the maximum path length of their pathways, and the scoring metric) to maximize high-scoring pathways in the real PPI network and minimized those in the randomized networks. They chose to search for paths of length at most 8 because the average shortest path length between any two proteins in their PPI graph was 7.4, and because a fraction of pathways with high microarray clustering ratios over various shortest path lengths peaked at 8. Their method reproduced many of the essential elements of the pheromone response, cell wall integrity, and filamentation MAPK pathways, but it failed to model the High Osmolarity (HOG) MAPK pathway due to missing interactions (false-negatives) in the PPI network.

We addressed the issue of identifying linear pathways in a yeast PPI network [132]. We focused on finding and exploiting the basic structure these pathways exhibit in PPI networks. We used MAPK as our model pathways, because they are among the most thoroughly studied networks in yeast and because they exhibit linearity in structure [136]. These pathways had source and sink nodes of low degree and internal nodes of high degree in the yeast PPI network. Thus, we used the following model to extract linear pathways from a PPI network: we constructed a shortest path from a transmembrane or sensing protein of low degree to a transcription factor protein of low degree, such that the internal nodes on the shortest path are of high degree; we also included high degree first and second neighbors of internal nodes of such a shortest path into these predicted pathways (for more details see [132]). In this way we extracted 399 predicted pathways (an example of a predicted pathway is presented in Figure 8).

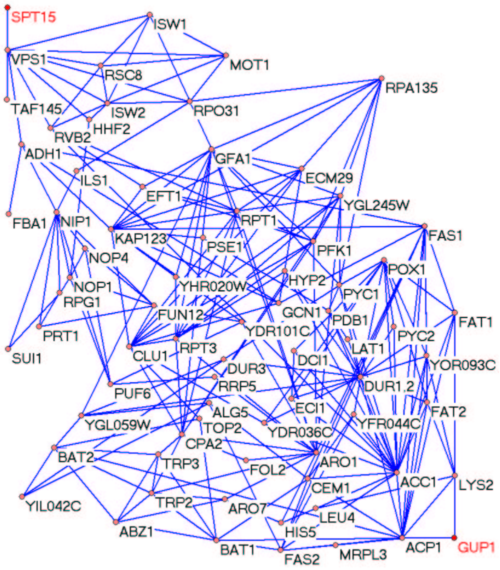

Figure 8: An example of a predicted pathway [132]. Note that this predicted pathway is presented as a subgraph of the PPI graph, and thus some of its internal vertices appear to be of low degree, even though they have many more interactions with proteins outside of this predicted pathway in the PPI graph.

Other theoretical approaches have been taken to model pathways. They involve system stoichiometry, thermodynamics etc. (for example, see [138]). Also, methods for extraction of pathway information from on-line literature are being developed [125] [62] [93]. Such approaches are beyond the scope of our attention, and thus are not presented here.

3.4 Properties of PPI Networks

Jeong, Tombor, Mason, Albert, Oltvai, and Barabasi have studied properties of PPI networks [82] [80]. They first studied metabolic pathway networks of different organisms from the WIT database [126]. This database contains predicted pathways based on the annotated genome of an organism and data established from the biochemical literature. They have shown that metabolic networks of 43 organisms from the WIT database, containing 6 archaea, 32 bacteria, and 5 eukaryota, all have scale-free topology with $P(k) \approx$

$k^{-2.2}$ for both in- and out-degrees [82]. The diameter of the metabolic networks was the same for all 43 organisms, indicating that with increasing organism complexity, nodes are increasingly connected. A few hubs dominated these networks, and upon the sequential removal of the most connected nodes, the diameter of the network rose sharply. Only around 4% of the nodes were present in all species, and these were the ones that were most highly connected in any individual organism; species-specific differences among organisms emerged for less connected nodes. A potential bias introduced by a high interest and research being done on some and a lack of interest and research being done on other proteins may also have contributed to this effect. In addition, when they randomly removed nodes from these networks, the average shortest path lengths did not change indicating insensitivity to random errors in these networks.

In their later study, Jeong, Mason, Barabasi, and Oltvai [80] analyzed the *S. cerevisiae* PPI network constructed on $1,870$ proteins and $2,240$ interactions derived from the yeast two-hybrid study of Uetz *et al.* [156] and DIP database [167]. They determined that the yeast PPI network and the PPI network of the human gastric pathogen *Helicobacter pylori* [133] also have heterogeneous scale-free network topology with a few highly connected proteins and numerous less connected proteins. They ordered all proteins in the yeast PPI graph according to their degree and examined the correlation between removal of a protein of a certain degree and the lethality of the organism. They demonstrated the same tolerance to random errors, coupled with fragility against the removal of high-degree nodes as in the metabolic networks: even though about 93% of proteins had degree at most 5, only about 21% of them were essential; on the other hand, only 0.7% of the yeast proteins with known phenotypic profiles had degree at least 15, but 62% of them were essential. They concluded that there has been evolutionary selection of a common large-scale structure of biological networks and hypothesized that future systematic PPI network studies in other organisms will uncover an identical PPI network topology. Our results on a larger yeast PPI network confirm their hypothesis [132].

Maslov and Sneppen [98] studied two networks, the protein interaction network derived from two-hybrid screens of Ito *et al.* [76], and the genetic regulatory network from YPD. A *genetic regulatory network* of a cell is formed by all pairs of proteins in which one protein directly regulates the abundance of the other. In most of these networks regulation happens at the transcription level, where a transcription factor regulates the RNA transcription of the controlled protein. These networks are naturally directed. Maslov and Sneppen established that the degree distribution of the PPI network on $2,378$ yeast proteins and $4,549$ interactions followed power law $\frac{1}{k^{2.5\pm0.3}}$, where k was between 2 and about 100. Their genetic regulatory network consisted of 682 proteins and $1,289$ edges. Both networks had a small number of high-degree nodes (hubs). The main contribution of this paper is the demonstration that both the interaction and the regulatory network had edges between hubs systematically suppressed, while those between a hub and a low-connected protein were favored. Furthermore, they demonstrated that hubs tend to share few of their neighbors with other hubs. They hypothesized that these effects decrease the likelihood of “cross talk” between different functional modules of the cell and increase the overall robustness of a network by localizing effects of harmful perturbations. They offer this as an explanation of why the correlation between the connectivity of a given protein and the lethality of the mutant cell lacking this protein was not very strong [80]. We offer an alternative explanation to this phenomenon [132]. We showed that hubs whose removal disconnects the PPI graph are likely to cause lethality (see below).

The PPI network we analyzed consisted of the top $11,000$ interactions among $2,401$ proteins from the study of von Mering *et al.* [158], which utilizes high confidence interactions detected by diverse experimental methods [132]. We confirmed previously noted result on smaller networks [80] demonstrating that *viable* proteins, whose disruption is non-lethal, have a degree that is half that of *lethal* proteins, whose mutation causes lethality (see Figure 9); proteins participating in *genetic interaction pairs* in the PPI network, i.e., combinations of non-lethal mutations which together lead to lethality or dosage lethality, appeared to have degree closer to that of viable proteins [132]. In this PPI network, lethal proteins were more frequent in the top 3% of high degree nodes compared to viable ones, while viable mutations were more frequent amongst the nodes of degree 1. Interestingly, lethal mutations were not only highly connected nodes within the network, but were nodes whose removal caused a disruption in network structure – it disconnected one part of the network from the other. The obvious interpretation of these observations in the context of cellular wiring is that lethality can be conceptualized as a point of disconnection in the PPI network. A contrasting property to hubs which are points of disconnection is the existence of alternative connections, called *siblings*, which covers nodes in a graph with the same neighborhood. We have observed that viable mutations have an increased frequency in the group of proteins that could be described as siblings within the network compared to lethal mutations or genetic interactions. This suggests the existence of alternate paths bypassing viable nodes in PPI networks, and offers an explanation why null mutation of these proteins is not lethal.

Ravasz, Somera, Mongru, Oltvai, and Barabasi have explored modular organization of metabolic networks [134]. They showed that metabolic networks of 43 organisms from the WIT database [82] [126] are all organized into many small, highly connected modules, which are combined in a hierarchical manner into larger units. Their motivation was the observed dichotomy between the two phenomena present in metabolic networks: on one hand are scale-free models of these networks with the observed power law degree distribution [18] [82] [159] and the existence of hubs which integrate all nodes into a single, integrated

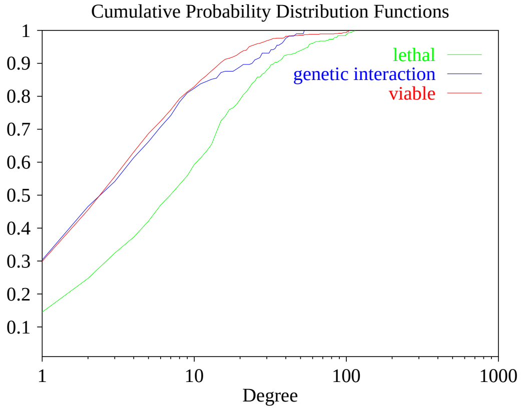

Figure 9: Cumulative distribution functions of degrees of lethal, genetic interaction, and viable protein groups in yeast PPI network constructed on $11,000$ interactions amongst $2,401$ proteins [132].

network, while on the other hand, these networks have high clustering coefficients [159] implying modular topologies. To examine this in detail they first determined the average clustering coefficients of metabolic networks of 43 different organisms, and established that all of them were an order of magnitude larger than expected for a scale-free network of similar size. This suggested high modularity of these networks. Also, the clustering coefficients of metabolic networks were independent of their sizes, contrasting the scale-free model, for which the clustering coefficient decreases as $n^{-0.75}$. To integrate these two seemingly contradicting phenomena, they proposed a heuristic model of metabolic organization which they call a "hierarchical" network. The construction is similar to the one described by Barabasi, Ravasz, and Vicsek [21] (presented in section 2), but a starting point in this network is a K_4 as a hypothetical module (rather than a P_3 [21]). They connect nodes of this starting module with nodes of three additional copies of K_4 so that the "central node" of the initial K_4 is connected to the three "external nodes" of new K_4s, as presented in Figure 10 (b). In this way they obtain a 16-node module. They repeat this process by making three additional copies of this 16-node module and connecting the "peripheral nodes" of the three new 16-node modules with the "central node" of the initial 16-node module (Figure 10 (c)). This process can be repeated indefinitely. They have shown that the architecture of this network integrates a scale-free topology with a modular structure; its power law degree distribution is $P(k) = k^{-2.26}$, which is in agreement with the observed $P(k) \approx k^{-2.2}$ [82], its clustering coefficient $C \approx 0.6$ is also comparable to the ones observed for metabolic networks, and most importantly, its clustering coefficient is independent of the network size. The hierarchical structure of this model is a feature that is not shared by the scale-free, or modular network models. They also demonstrated that for each of the 43 organisms, the clustering coefficient $C(k)$ of a degree k node is well approximated by $C(k) \approx k^{-1}$, which is in agreement with the Dorogovtsev, Goltsev, and Mendes theoretical result presented in section 2.4 establishing that the clustering coefficient of a degree k node of a scale-free network follows the scaling law $C(k) \approx k^{-1}$ [47]. Thus, their hierarchical network model includes all the observed properties of metabolic networks: the scale-free topology, the high, system size independent clustering coefficient, and the power law scaling of $C(k)$. To inspect whether their model reflects the true functional organization of cellular metabolism, they focused on the extensively studied metabolic network of *E. coli*, and established that their model closely overlaps with *E. coli*'s known metabolic network. They hypothesized that this network architecture may be generic to cellular organization networks. The existence of small, highly frequent subgraphs in these networks, called "network motifs" (described below) [143] [110] makes this hypothesis even more plausible.

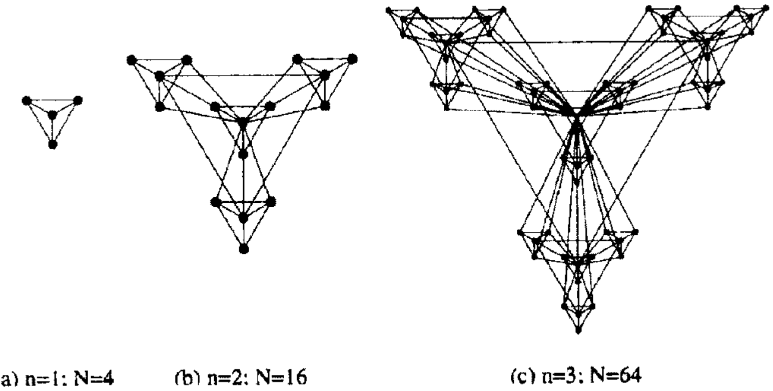

Figure 10: Three steps in the construction of a hierarchical model network. Taken from [134].

Jeong, Barabasi, Tombor, and Oltvai [79] have further extended these results by applying the same analysis on the complete biochemical reaction network of the 43 organisms from the WIT database. They obtained these networks by combining all pathways deposited in the WIT database for each organism into a single network. These networks are again naturally directed, so they examined their in- and out-degree distributions. All of the 43 networks obtained in this way exhibited a power-law distribution for both in- and out-degrees, from which they concluded that scale-free topology is a generic structural organization of the total biochemical reaction networks in all organisms in all three domains of life. However, the largest portion of the WIT database consists of the data on core metabolism pathways, followed by the data on information transfer pathways, so their results may have largely been influenced by the domination of metabolic pathways. To resolve this issue, they performed the same analysis on the information transfer pathways alone, since apart from the metabolic pathways, these were the only ones present in high enough quantities for doing statistical analyses. The analysis of the information transfer pathways of 39 organisms (four of the 43 organisms had their information pathways of too small size for doing statistics) revealed the same power-law degree distribution both for in- and out-degree as seen for metabolic and complete biochemical reaction networks. Similarly, they confirmed that the network diameter (which they defined as the average of shortest path lengths between each pair of nodes) remained constant and around 3 for biochemical reaction networks, metabolic networks, and information transfer networks of all 43 organisms, irrespective of the network sizes. Thus, in these networks, the average degree of a node increases with the network size. This is contrary to the results on real non-biological networks, in which the average degree of a node is fixed, so the diameter of the network increases logarithmically with the network size [18] [163] [24]. They also found that only about 5% of all nodes were common to the biochemical reaction networks of all 43 species, and that these were the highest degree nodes. They obtained the same result when they repeated this analysis for metabolic and information transfer networks alone.

Barabasi, Dezso, Ravasz, Yook, and Oltvai [20] also described a minor variation of their hierarchical network model (presented in Figure 11), and showed that four independent yeast PPI networks derived from the DIP database [168], the study of Ito *et al.* [76], the MIPS database [108], and the study of Uetz *et al.* [156], all had hierarchical structures with $C(k)$ scaling as k^{-1}.

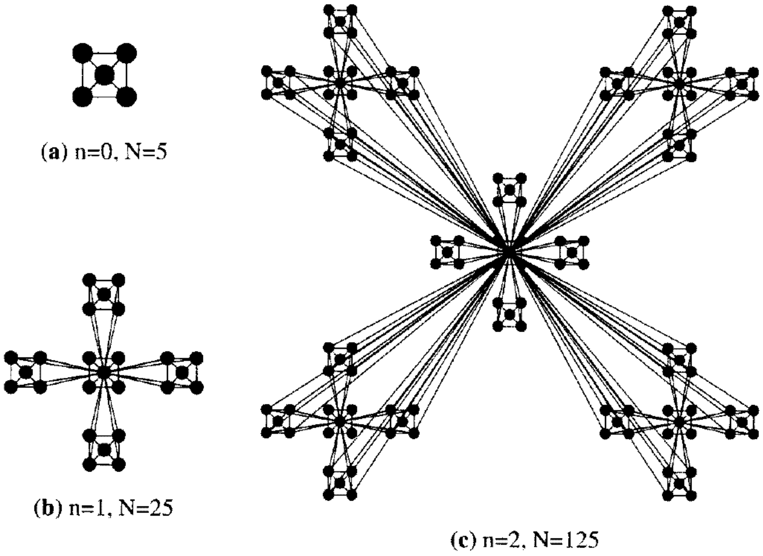

Figure 11: Three steps in the construction of a hierarchical model network. Taken from [20].

Shen-Orr, Milo, Mangan, and Alon analyzed the transcriptional regulation network of *Escherichia coli* and defined "network motifs" as patterns of interconnections that recur in many different parts of a network at frequencies much higher than those found in randomized networks [143]. They represented the transcriptional network as a directed graph in which each node represents an *operon*, a group of contiguous genes that are transcribed into a single m-RNA molecule, and each edge is directed from an operon that encodes a transcription factor to an operon that is regulated by that transcription factor. They discovered that much of the network is composed of repeated appearances of three highly significant motifs each of which has a specific function in determining gene expression. These three motifs they called “feedforward loop”, “single-input module” (SIM), and “dense overlapping regulons” (DOR) (a *regulon* stands for a group of coordinately regulated operons). An illustration of these three motifs is presented in Figure 12. They used straightforward adjacency matrix manipulation algorithms to detect motifs on 3 and 4 nodes and to detect SIMs; they defined a non-metric distance measure between operons and used a standard average-linkage clustering algorithm [49] to identify DORs. They further described how feedforward loops can act as circuits that reject transient activation signals and respond only to persistent signals, while allowing a rapid system shutdown: X and Y act in an AND-gate-like manner to control operon Z. On the other hand, SIMs allow temporal ordering of activation of different genes with different activation thresholds, which is useful for processes that require several stages to complete, such as, for example, amino-acid biosynthesis processes. In addition to giving an explanation of how different processes work, the introduction of the above three motifs enabled Shen-Orr *et al.* to represent the E.coli transcriptional network in a compact, modular form. We hope that by performing analysis similar to the one described by Shen-Orr *et al.* we will be able to identify different motifs describing different functional protein groups in large, undirected PPI networks, which would mathematically describe cell processes, predict new processes, and aid in determining function of uncharacterized proteins (see section 5).

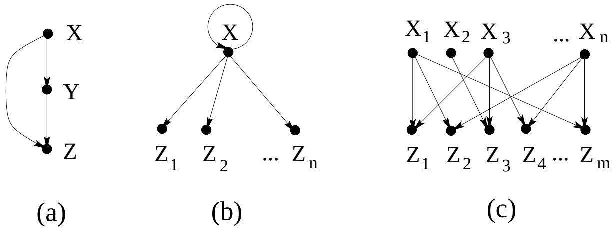

Figure 12: Motifs from [143]: (a) feedforward loop, (b) single input module (SIM), (c) dense overlapping regulons (DOR).

Milo, Shen-Orr, Itzkovitz, Kashtan, Chklovskii, and Alon further extended network motif analysis to different types of large networks and determined that different networks have different motifs [110]. They analyzed *E. coli* and *S. cerevisiae* gene regulation networks (transcription), the neuron connectivity network of *C. elegans*, seven food web networks, the ISCAS89 benchmark set of sequential logic electronic circuits, and a network of directed hyper-links between World Wide Web pages within a single domain. They searched for all possible 3- and 4-node directed subnetworks in these real large networks and compared the frequencies of occurrences of each of these small subnetworks in a real network with the frequencies of their occurrences in randomized networks that have the same connectivity properties and the same number of $(n-1)$-node subgraphs as the real networks, where n is the size of the motif they were trying to detect. They designed their random networks in this way in order to account for patterns that appear only because of the single-node characteristics of the network, such as the presence of nodes with a large degree, and also to ensure that a high significance was not assigned to a pattern only because it has a highly significant sub-pattern. They defined "network motifs" to be those patterns for which the probability of appearing in a randomized network an equal of greater number of times than in the real network is lower than 0.01. Thus, there may exist functionally important but not statistically significant patterns that their approach would miss. They further indicated that the number of appearances of each motif in the real networks appears to grow linearly with the system size, while it drops in their random networks; this drop is in accordance with an exact result on Erdos-Renyi random graphs in which the concentration C of a subgraph with n nodes and m edges (i.e., the fraction of times a given n-node subgraph occurs among the total number of occurrences of all possible n-node subgraphs) scales with network size S, as $C \approx S^{n-m-1}$ [31], which in the study of Milo *et al.* is equal to $\frac{1}{S}$, since all but one of their motifs have $n = m$. In addition, they established that the identified motifs are insensitive to data errors, since they do not change after addition, deletion, or rearrangement of 20% of the edges at random. They also tested their approach on an undirected yeast PPI network on $1,843$ nodes and $2,203$ edges [80] and identified one 3-node and one 4-node motif. They identified two 4-node "anti-motifs", the patterns whose probability of appearing in randomized networks fewer times than in the real network is less than 0.01, and $N_{rand} - N_{real} > 0.1N_{rand}$, where N_{rand} and N_{real} are the number or subgraph appearances in a real and in randomized networks respectively.

We investigated if distinct functional classes of proteins have differing network properties [132]. Our results support the findings that complex networks comprise simple building blocks [143] [110]. Since different building blocks and modules have different properties, it can be expected that they serve different functions. To examine this, we used the functional classifications in the MIPS database [108] to statistically determine graph properties for each group. We observed that proteins involved in translation appear to have the highest average degree, while transport and sensing proteins have the lowest average degree. Figures 13 A and 13 B support this result as half of the nodes with degrees in the top 3% of all node degrees are translation proteins, while none belong to amino-acid metabolism, energy production, stress and defense, transcriptional control, or transport and sensing proteins. This is further supported by the observation that metabolic networks across 43 organisms tested have an average degree of < 4 [82]. By intersecting each of the lethal, genetic interaction, and viable protein sets with each of the functional groups, we observed that amino-acid metabolism, energy production, stress and defense, transport and sensing proteins are less likely to be lethal mutations (see Figure 13 C). Of all functional groups, transcription proteins have the largest presence in the set of lethal nodes on the PPI graph; approximately 27% of lethals on the PPI graph are transcription proteins, as illustrated in Figure 13 C. Notably, amongst all functional groups, cellular organization proteins have the largest presence in hub nodes whose removal disconnects the network (the nodes whose removal disconnects the network we called *articulation points*; see Figure 13 D).

We also constructed a simple model for predicting new genetic interaction pairs in the yeast PPI network [132]. This model is based on the distribution of shortest path lengths between known genetic interaction pairs in the PPI network. In addition, we suggested a way to extract "bottle neck" proteins form PPI networks which are likely to be important proteins in these networks: 7 out of the top 10 bottle neck proteins were inviable and structural proteins in the yeast PPI network.

New approaches integrating different high-throughput methods in order to describe known and to predict new biological phenomena have started to appear. Since most high-throughput techniques contain noise, they may complement each other yielding less noisy data. One approach in this direction is integrating graph theoretic PPI analysis with the results of microarray experiments (for example, see [81]). Even though they are very interesting, these approaches are currently out of the scope of our attention, and thus we do not survey them in this article. However, in the Future Research section (section 5), we propose an integrated approach for the construction and analysis of putative PPI networks for different organisms.

4 Detection of Dense Subnetworks

From the above we can see that even though currently available PPI networks contain high degree of false positives and false negatives, they do have structure. One of our goals is to discover more PPI network structure and ultimately exploit it for designing efficient, robust, and reliable algorithms for extracting graph substructures embedded in these networks that have biological meaning and describe biological processes. Our previous discussion suggests that one example of such substructures may be dense subgraphs of these networks representing core proteins of protein complexes. Thus, we present here some of the recent graph theoretic techniques that could be used as a first step towards addressing extraction of these dense subgraphs in PPI graphs. It is possible, however, that protein complexes (as well as pathways) have distinct graph theoretic structures requiring novel graph theoretic approaches for their detection in PPI networks.

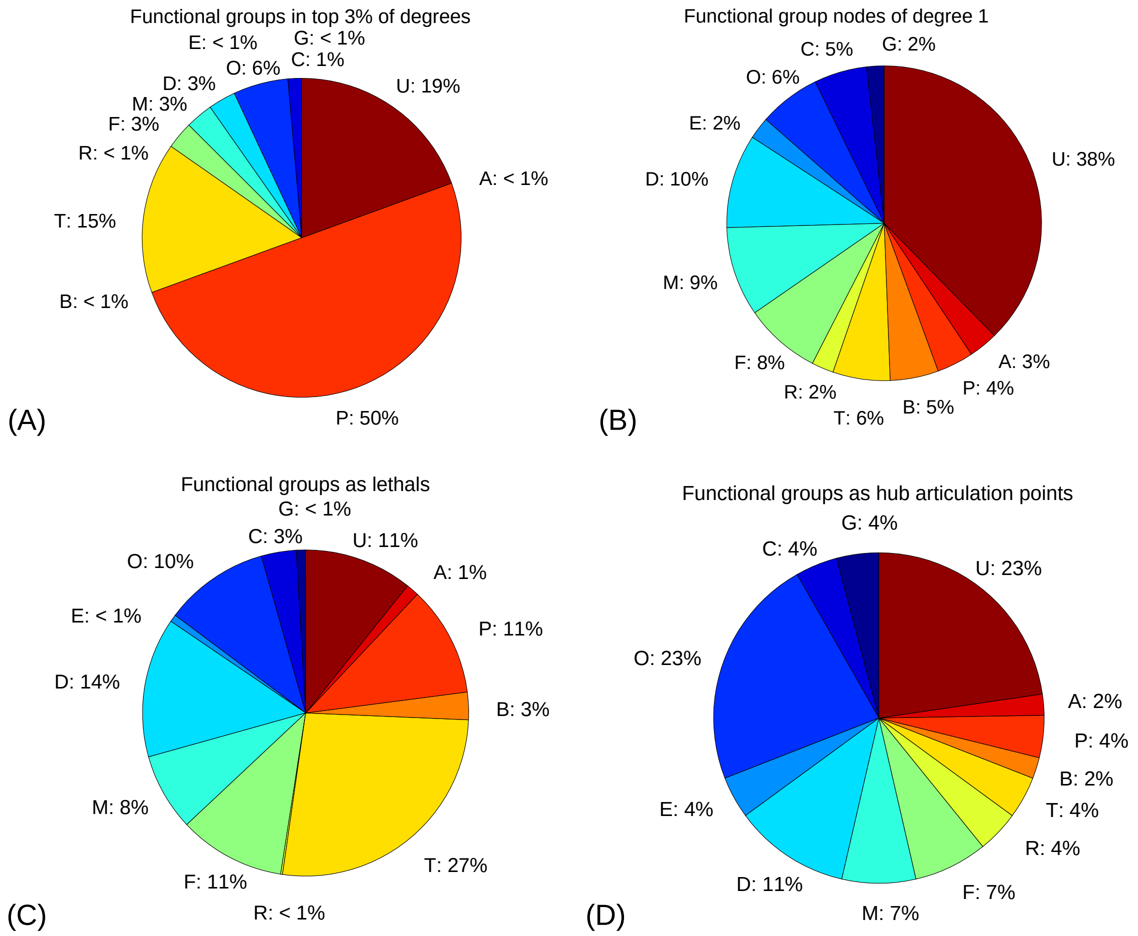

Figure 13: Pie charts for functional groups in the PPI graph: G – amino acid metabolism, C – cellular fate/organization, O – cellular organization, E – energy production, D – genome maintenance, M – other metabolism, F – protein fate, R – stress and defense, T – transcription, B – transcriptional control, P – translation, A – transport and sensing, U – uncharacterized. **A.** Division of the group of nodes with degrees in the top 3% of all node degrees. **B.** Division of nodes of degree 1. Compared with Figure 13 A, translation proteins are about 12 times less frequent, transcription about 2 times, while cellular fate/organization are 5 times more frequent, and genome maintenance, protein fate, and other metabolism are about 3 times more frequent; also, there are twice as many uncharacterized proteins. **C.** Division of lethal nodes. **D.** Division of articulation points which are hubs.

Clustering is an important problem in many disciplines including computational biology. Its goal is to partition a set of elements into subsets called *clusters* so that the elements of the same cluster are similar to each other (this property is called *homogeneity*) and elements from different clusters are not similar to each other (this property is called *separation*). Homogeneity and separation can be defined in many different ways leading to different optimization problems. Elements belonging to the same cluster are usually called *mates* and the elements belonging to different clusters are called *non-mates*. Clustering problems and algorithms can often be expressed in graph-theoretic terms. For example, a *similarity graph* can be constructed so that nodes represent elements and edge weights represent similarity values of the corresponding elements. In the analysis of PPI networks, weights on edges have not yet been incorporated in the model, but it may be useful to incorporate them to represent the confidence (in the von Mering sense [158]) that the two proteins really interact, or the strength of the interaction.

Several graph theoretic techniques have been developed recently to cluster microarray gene expression profiles. Shamir and Sharan gave a good overview of these and other gene expression clustering methods [141]. Enright has addressed clustering of protein sequences [53]. We describe here those graph theoretic methods that may be used for identifying protein complexes, that is, "dense" subgraphs, of PPI networks. We do not attempt to give an exhaustive survey of this large research area, but rather, point to some of the techniques as illustrations of the mathematical arsenal available for attacking the problem of protein complex identification in PPI networks. Several of the dense subgraph identification methods that we survey were first developed as "exact" algorithms with proven properties in terms of solution quality and time complexity, and later were modified to include heuristics which make them more efficient [70] [142]. Several of them have a probabilistic nature [27] [157] [53].

To give a brief historical overview, we mention some of the important early works in the area of graph theoretic clustering. To narrow the scope, we only present the results relevant to the recent graph clustering algorithms used in computational biology that we describe below. We follow the background sections of Hartuv and Shamir [70] and van Dongen [157], whose results we describe in more detail later in this section.

Matula addressed the problem of graph theoretic clustering in a series of papers [99] [100] [101] [102]. He observed that highly connected regions of similarity graphs are useful in cluster analysis. He defined the *cohesiveness function* for every vertex and edge in a graph to be the maximum edge-connectivity of any subgraph containing that vertex/edge. By deleting all elements of a graph of cohesiveness less than k, he obtained maximal k-edge-connected subgraphs of the graph. He first identified clusters by using a constant k [100], and later modified the technique to obtain, for any k, clusters which are maximal k-edge-connected subgraphs that do not contain a subgraph with higher connectivity [101]. He presented several graph cluster concepts [102] using the subgraph notions of: k-bond – a maximal subgraph S such that every node in S has degree at least k in S; k-component – a maximal subgraph S such that every pair of nodes in S is joined by k edge-disjoint paths in S; k-block – a maximal subgraph S such that every pair of nodes in S is joined by k vertex-disjoint paths in S. These notions imply cluster methods which are successive refinements going from bond to component to block. These algorithms require solving minimum cut network flow problem and their time complexities are at least cubic in the input graph vertex set size for connected graphs. Hartuv, Shamir, and Sharan [69] [70] [142] [140] have built on Matula's work producing faster exact graph clustering algorithms and also introducing heuristics to further speed up their algorithms (see below). Alpert and Kahng give a good survey of other graph theoretic clustering techniques, including the probabilistic ones [8]. We now turn to describing some of the recent graph theoretic clustering techniques that have successfully been used in biological applications.

The Highly Connected Subgraph (HCS) [69] [70] and CLuster Identification via Connectivity Kernels (CLICK) [142] [140] algorithms operate on a similar principle. The input is a similarity graph, and the algorithm first considers if the graph satisfies a stopping criterion, in which case it is declared a "kernel". Otherwise, the graph is partitioned into two subgraphs, separated by a minimum weight edge cut, and the algorithm recursively proceeds on the two subgraphs, outputting in the end a list of kernels that represent a basis for the possible clusters. The overview of this general algorithm scheme is presented in Algorithm 1 (adapted from [141]). HCS and CLICK construct similarity graphs differently and have different stopping criteria. We now describe their distinguishing basic properties.

Algorithm 1: FORM-KERNELS(G)

```
if $V(G) = \{v\}$ then
    move $v$ to the singleton set
end
else
    if $G$ is a kernel then
        output $V(G)$
    end
end
else
    $(H, \overline{H}) \leftarrow MinWeightEdgeCut(G)$;
    Form-Kernels($H$);
    Form-Kernels($\overline{H}$);
end
```

The input into the HCS is an unweighted similarity graph G. A *highly connected subgraph (HCS)* is defined to be an induced subgraph H of G such that the number of edges in a minimum edge cut of H is bigger than $\frac{|V(H)|}{2}$. That is, if any $\lfloor \frac{|V(H)|}{2} \rfloor$ of edges of H are removed, H remains connected. The algorithm uses these highly connected subgraphs as kernels. Hartuv and Shamir proved that this algorithm produces homogeneous and well separated clusters [69]. Clusters are homogeneous, since the diameter of each cluster is at most 2 and each cluster is at least half as dense as a clique. They are well separated, since any non-trivial split by the algorithm happens on subgraphs that are likely to be of diameter at least 3. The running time of the HCS algorithm is bounded by $2N \times f(n,m)$, where N is the number of clusters found (often $N \ll n$) and $f(n,m)$ is the time complexity of computing a minimum edge cut of a graph with n nodes and m edges. Currently the fastest deterministic minimum edge cut algorithms for unweighted graphs are of time complexity $O(nm)$ and are due to Matula [103] and Nagamochi and Ibaraki [114]. The fastest simple deterministic minimum edge cut algorithm for weighted graphs is of time complexity $O(nm + n^2 \log n)$ and is due to Stoer and Wagner [150]; it is implemented by Mehlhorn and is part of the Leda library [107]. Several heuristics are used to speed up the HCS algorithm in practice. The first one is called *Iterated HCS* and is based on the fact that HCS arbitrarily chooses a minimum edge cut when the same minimum cut value is obtained by several different cuts. This process will often break small clusters into singletons. To avoid this, several iterations of HCS could be performed until no new cluster is found. This would theoretically add another $O(n)$ factor to the running time, but in practice only between 1 and 5 iterations are usually needed. Another heuristic is called *Singletons Adoption* and is based on the principle that singleton vertices get “adopted” by clusters based on their similarity to the clusters. For each singleton node x, the number of neighbors of x in each cluster as well as in the set of all singletons $\mathcal{S}$ is computed, and x is added to a cluster (never to $\mathcal{S}$) with the maximum number of neighbors $\mathcal{N}$, if $\mathcal{N}$ is sufficiently large. This process is repeated a specified number of times to account for changes in clusters resulting from previous adoptions. The last HCS algorithm heuristic described by Hartuv and Shamir is based on *removing low degree vertices*. This is done to speed up the algorithm, since if the input graph contains many low degree vertices, one iteration of the minimum edge cut algorithm may only separate a low degree vertex from the rest of the graph contributing to increased computational cost at a low informative value in terms of clustering. This is especially expensive for large graphs with many low degree vertices. For example, around 28% of the vertices of the PPI graph constructed on the top $11,000$ interactions (and $2,401$ proteins) from the study of von Mering *et al.* [158], and around 13% of the vertices of the PPI graph constructed on all ≈ 78K of the yeast protein-protein interactions (and $5,321$ proteins) [158] are of degree 1, so this heuristic may significantly speed up the HCS algorithm applied to these data sets. We implemented the HCS algorithm without any heuristics and applied it to several PPI graphs constructed on the data set of von Mering *et al.* [158], as described in section 3.3.1. Our results show that clusters identified this way have high overlap with known MIPS protein complexes and a much higher functional group homogeneity than expected at random [132] (also see section 3.3.1). Thus, high precision is favored by this method of protein complex identification; in contrast, Bader and Hogue’s approach [15] improves recall at the expense of precision.

The CLICK algorithm [140] builds on Hartuv and Shamir’s HCS algorithm [69] by incorporating statistical techniques to identify kernels. Similar to HCS, a weighted similarity input graph is recursively partitioned into components using minimum weight edge cut computations. The edge weights and the stopping criterion of the recursion have probabilistic meaning. Pairwise similarity values between mates are assumed to be normally distributed with mean μ_T and variance σ_T, and pairwise similarity values between non-mates are assumed to be normally distributed with mean μ_F and variance σ_F, where $\mu_T > \mu_F$ (this is observed on real data and can also be theoretically justified [142]). Also, the probability p_{mates} of two randomly chosen elements being mates is taken into account when computing edge weights. If the input similarity matrix between elements is denoted by $S = (S_{ij})$, the weight of an edge (i, j) in the similarity graph is computed as $w_{ij} = \ln \frac{Prob(i,j \text{ are mates}|S_{ij})}{Prob(i,j \text{ are non-mates}|S_{ij})} = \ln \frac{p_{mates} f(S_{ij}|i,j \text{ are mates})}{(1-p_{mates}) f(S_{ij}|i,j \text{ are non-mates})}$, where $f(S_{ij}|i,j \text{ are mates}) = \frac{1}{\sqrt{2\pi}\sigma_T} e^{-\frac{(s_{ij}-\mu_T)^2}{2\sigma_T^2}}$ and $f(S_{ij}|i,j \text{ are non-mates}) = \frac{1}{\sqrt{2\pi}\sigma_F} e^{-\frac{(s_{ij}-\mu_F)^2}{2\sigma_F^2}}$, and thus $w_{ij} = \ln \frac{p_{mates}\sigma_F}{(1-p_{mates})\sigma_T} + \frac{(S_{ij}-\mu_F)^2}{2\sigma_F^2} - \frac{(S_{ij}-\mu_T)^2}{2\sigma_T^2}$. To increase efficiency, they omit from the graph edges whose weight is below a predefined non-negative threshold. They determine kernels in the following way. They call a connected subgraph G *pure*, if $V(G)$ contains only elements of some cluster. For each cut C of a connected graph, they test the following hypotheses:

H_0^C : C contains only edges between non-mates.

H_1^C : C contains only edges between mates.

G is pure if and only if H_1^C is accepted for every cut C of the graph G; in this case they say that G is a kernel. If G is not a kernel, they partition it along a cut C for which the ratio $Pr(H_1^C|C)/Pr(H_0^C|C)$ is minimum. They expand kernels obtained in this way to obtain clusters, first by singleton adoptions, then by merging "similar" clusters, and finally, by performing another round of singleton adoptions. For more details, see Algorithm 2 and [142].

Algorithm 2: CLICK(G)

```
Singletons $\mathcal{S} \leftarrow$ complete set of elements $N$;
while some change occurs do
    Execute FORM-KERNELS($G(\mathcal{S})$);
    Let $\mathcal{K}$ be the list of produced kernels;
    Let $\mathcal{S}$ be the set of singletons produced;
    Adoption($\mathcal{K}, \mathcal{S}$)
end
Merge($\mathcal{K}$);
Adoption($\mathcal{K}, \mathcal{S}$)
```

To speed up the algorithm, they used the following heuristics. Similar to removing low degree nodes for HCS, they screen for low weight nodes (the weight of a node v is the sum of weights of the edges incident on v) from large components in the following way. They first compute the average node weight W of the component and multiply W by a factor proportional to the logarithm of the component size; the result is denoted by W^*. Nodes with weight less than W^* are removed repeatedly, updating the weight of the remaining nodes each time a node is removed, until the updated weight of all remaining nodes is greater than W^*. The removed nodes are added to the singleton set and handled later. The second heuristic they used is the following. Instead of finding computationally expensive minimum weight edge cuts (they used Hao and Orlin's [68] $O(n^2\sqrt{m})$ algorithm that has been shown to outperform other minimum weight edge cut algorithms in practice [40]), they computed a minimum $s - t$ cut of the underlying unweighted graph using Dinic's $O(nm^{2/3})$ time algorithm [59], with s and t chosen to be nodes that achieve the diameter d of the graph, when $d \geq 4$ (they used the $O(n+m)$ time breadth first search algorithm to find the diameter of the graph).

Ben-Dor, Shamir, and Yakhini [27] developed a polynomial time algorithm for finding the clustering with high probability under the following stochastic model of the data. They assume that the correct structure of the input graph is a disjoint union of cliques (cliques represent clusters), but that errors were introduced to it by independently adding or removing edges with probability $\alpha < \frac{1}{2}$. Their heuristic Cluster Affinity Search Technique (CAST) algorithm is built on their theoretical Parallel Classification with Cores (PCC) algorithm which solves the problem to a desired accuracy with high probability in time $O(n^2(\log n)^c)$ (for more details see [27]). The input to CAST is the similarity matrix S. CAST uses the notion of the *affinity* of an element v to a putative cluster C, $a(v) = \sum_{i \in C} S(i,v)$, and the affinity threshold parameter t. It generates clusters sequentially by starting with a single element and adding or removing elements from a cluster if their affinity is larger or lower than t, respectively. This process is repeated until it stabilizes. The details are shown in Algorithm 3. In the end, an additional heuristic tries to ensure that each element has the affinity to its assigned cluster higher than to any other cluster by moving elements until the process converges, or some maximum number of iterations is completed.

Algorithm 3: CAST(S)

```
while there are unclustered elements do
    Pick an unclustered element to start a new cluster $C$;
    repeat
        add an unclustered element $v$ with maximum affinity to $C$, if $a(v) > t|C|$;
        remove an element $u$ from $C$ with minimum affinity, if $a(u) \leq t|C|$;
    until no changes occur;
    Add $C$ to the list of final clusters;
end
```

Recently, van Dongen [157] developed a new clustering algorithm which was later used to cluster protein sequences into families [53] [55]. The algorithm is called the Markov Cluster (MCL) algorithm and it was designed to cluster undirected unweighted and weighted graphs. The algorithm simulates flow within a graph, promoting flow where the current is strong and demoting flow where the current is weak until the current across borders between different groups of nodes withers away revealing a cluster structure of the graph (an illustration is presented in Figure 14).

More formally, the MCL algorithm deterministically computes the probabilities of random walks through

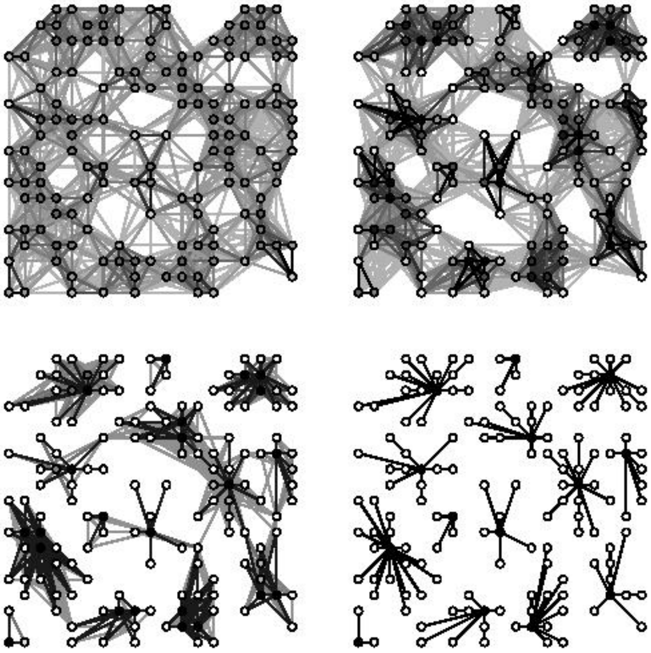

Figure 14: Stages of flow simulation by the MCL process. Taken from [157].

the graph and uses two operators, expansion and inflation, to transform one set of probabilities into another. It uses stochastic matrices (also called Markov matrices) that capture the mathematical concept of random walks on a graph. Following the notation of van Dongen [157], for a weighted directed graph $G = (V, E)$, with $|V| = n$, its *associated matrix* M_G is an $n \times n$ matrix with entries $(M_G)_{pq}(1 \leq p, q \leq n)$ being equal to weights of edges between vertices p and q (clearly, weights of all edges of an unweighted graph are equal to 1). Similarly, every square matrix M can be assigned an *associated graph* G_M. For a graph G on n nodes and its associated matrix $M = M_G$, the *Markov matrix* associated with G, denoted by $\mathcal{T}_G$, is obtained by normalizing each column of M so that it sums to 1, i.e., if D is a diagonal matrix with $D_{kk} = \sum_i M_{ik}$ and $D_{ij} = 0$ for $i \neq j$, then $\mathcal{T}_G$ is defined as $\mathcal{T}_G = M_G D^{-1}$. A column j of $\mathcal{T}_G$ corresponds with node j of the stochastic graph associated with $\mathcal{T}_G$, and the matrix entry $(\mathcal{T}_G)_{ij}$ corresponds to the probability of going from node j to node i. Given such a matrix $\mathcal{T}_G \in \mathbb{R}^{n \times n}, \mathcal{T}_G \geq 0$, and a real number $r > 1$, let $\Gamma_r : \mathbb{R}^{k \times k} \to \mathbb{R}^{k \times k}$ be defined as $(\Gamma_r \mathcal{T}_G)_{pq} = ((\mathcal{T}_G)_{pq})^r / \sum_{i=1}^{n}((\mathcal{T}_G)_{iq})^r$. Γ_r is called the *inflation* operator with power coefficient r and the Markov matrix resulting from inflating each of the columns of $\mathcal{T}_G$ with power coefficient r is written as $\Gamma_r \mathcal{T}_G$. For $r > 1$, inflation changes the probabilities associated with the collection of random walks departing from a node (corresponding to a matrix column) by favoring more probable walks over less probable ones. Inflation can be altered by changing r: larger r makes inflation stronger and produces “tighter” clusters. *Expansion* corresponds with computing “longer” random walks. It associates new probabilities with all pairs of nodes with one node being the point of departure and the other being the destination. It relies on the observation that longer paths are more common within clusters than between different clusters, and thus the probabilities associated with node pairs which are within the same cluster will, in general, be relatively large, since there are many ways of going from one node to the other. Expansion is achieved via matrix multiplication. The MCL algorithm iterates the process of expanding information flow via matrix multiplication and contracting it via inflation. The basics of the MCL algorithm are presented in Algorithm 4.

Algorithm 4: $\mathrm{MCL}(G, \Delta, e_{(i)}, r_{(i)})$

```
    # $G$ is a graph with every node of degree $\geq 1$;
    # $e_{(i)}$ is a sequence of $e_i \in \mathbb{N}, e_i > 1, i = 1, \ldots$;
    # $r_{(i)}$ is a sequence of $r_i \in \mathbb{R}, r_i > 0, i = 1, \ldots$;
    $G = G + \Delta$; # Possibly add (weighted) self-loops;
    $T_1 = \mathcal{T}_G$;
    $k = 0$;
    repeat
        $k = k + 1$;
        $T_{2k} = (T_{2k-1})^{e_k}$; # Expansion;
        $T_{2k+1} = \Gamma_{r_k}(T_{2k})$; # Inflation;
    until $T_{2k+1}$ is (near-) idempotent;
    Interpret $T_{2k+1}$ as a clustering.
```

Iterating expansion and inflation results in the matrix that is idempotent under both matrix multiplication and the inflation (such a matrix is called *doubly idempotent*), that is, an equilibrium state is reached when a matrix does not change with further expansion and inflation. The graph associated with such a matrix consists of different directed connected star-like components with an attractor in the centre (see bottom right picture in Figure 14). Each of these components is interpreted as a cluster. Theoretically, there may exist nodes connected to different stars, which is interpreted as cluster overlap [157]. The algorithm iterants converge nearly always to the doubly idempotent matrix. In practice they start noticeably converging after 3 to 10 iterations. Van Dongen proved quadratic convergence of the MCL process in the neighborhood of doubly idempotent matrices [157]. The row of expansion powers, $e_{(i)}$, and the row of inflation powers, $r_{(i)}$, in Algorithm 4 influence the granularity of the resulting clustering.

As mentioned above, Enright, van Dongen, and Ouzounis [55] used the MCL algorithm to cluster protein sequences into families. For this purpose, nodes of the graph represented proteins, edges represented sequence similarities between the corresponding proteins, and edge weights corresponded to sequence similarity scores obtained from an algorithm such as BLAST [9] [10]. The overview of their algorithm, called Tribe-MCL, is presented in Algorithm 5. Tribe-MCL allowed hundreds of thousands of sequences to be accurately classified in a matter of minutes [53].

Algorithm 5: TRIBE-MCL(SET OF PROTEIN SEQUENCES S)

```
    All versus all BLAST($S$);
    Parse results and symmetrify similarity scores;
    Produce similarity matrix $M$;
    Transform $M$ to normalize similarity scores ($-\log$(E-value)) and generate transition probabilities;
    MCL($G_M$);
    Post process and correct domains of resulting protein clusters (families).
```

5 Future Research

From the above we can see that the analysis of PPI networks is a young, multidisciplinary research area with many open problems. We emphasize here those open problems that we consider the most interesting.

Understanding interactions between proteins in a cell may benefit from a better model of a PPI network. A full description of protein interaction networks requires a model that would encompass the undirected physical protein-protein interactions, other types of interactions, interaction confidence level, source (or method) and multiplicity of an interaction, directional pathway information, temporal information on the presence or absence of a PPI, information on the strength of the interactions, and possibly protein complex information. This may be achieved by designing a weighting function and assigning weights to nodes and edges of a PPI network to incorporate temporal and other interaction specific information, adding directionality to the network subgraphs, and building a hypergraph structure on the top of the network to incorporate information about complexes, or pathways in which proteins take part.

Another interesting research topic is finding an efficient and robust graph clustering algorithm that would reliably identify protein complexes, separate stable from transient complexes [78], or detect pathways in PPI networks, despite the noise present in PPI networks. Identifying graph theoretic structural properties that are common to protein complexes or certain pathway types in PPI networks may be crucial to designing such an algorithm. Similarly, modeling signaling pathways and finding efficient algorithms for their identification in PPI networks is another interesting topic. These algorithms would likely have to be stochastic, massively parallel, and use local search techniques, due to the presence of noise and large network sizes.

The existence of a “core proteome” has been hypothesized. It has been proposed that approximately 40% of yeast proteins are conserved through eukaryotic evolution [41]. We are approaching the moment when enough information would be available to verify the existence of such a proteome and discover its structural properties within PPI networks. It is already possible to take the first steps towards this goal with the currently available data. We propose to construct putative PPI networks for a number of eukaryotic organisms with mapped genomes by combining protein sequence similarities between different organisms with the known PPI networks of model organisms. With the set of putative PPI networks constructed in this way, it may be possible to do PPI network structural comparisons over different organisms. Preferential attachment to high degree nodes in real world networks has been suggested, implying that the core proteome would consist of high-degree nodes in PPI networks (described in previous sections). It is interesting to notice the discrepancy between the high degree of supposed “core proteome” proteins (hubs) and the separation of hubs by low-degree nodes noticed by Maslov and Sneppen [98]. Exploring the structural properties of this discrepancy may give an insight not only in the processes of evolution, but also in the properties that a better PPI network model should have. Research in this direction may result in construction of a stochastic, or deterministic large network model (similar to the model of Ravasz *et al.* [134] and the model of Jeong *et al.* [79] described above) which would provide a better framework for understanding PPI networks.

Other interesting topics for future research include distinguishing different graph theoretic properties of proteins belonging to different functional groups. Our results [132] suggest that such differences exist. One way to approach this problem would be to identify different network motifs (in the Shen-Orr [143], or some other sense) in the neighborhood of proteins belonging to the same functional group, and to compare the enrichment of these motifs over the functionally different sets of proteins. Along the same lines, it may be interesting to compare graph structures of the “same-function modules” over putative (or real, when they become available) PPI networks of different species and possibly infer common and differing elements in the structures of these modules. This could lead to construction of new models which could be used for identification of false positives and false negatives in PPI networks.

Integration of microarray data with PPI data may be beneficial for finding solutions to many of the above mentioned open problems.

Complex biological and artificial networks show graph-theoretic properties that reflect the function these networks carry [110] [170] [155] [166] [51] [66] [148]. A similar analysis could be applied to call graphs of large software [130] [131] [129]. A comparison between PPI networks, software call graphs, and other biological artificial networks may give further insight into the properties of large, evolving networks.

References

[1] J. Abello, A. Buchsbaum, and J. Westbrook. A functional approach to external graph algorithms. *Lecture Notes in Computer Science*, 1461:332–343, 1998.

[2] L. A. Adamic. The small world web. *Lecture Notes in Computer Science*, 1696:443–454, 1999.

[3] W. Aiello, F. Chung, and L. Lu. A random graph model for power law graphs. *Experimental Mathematics*, 10:53–66, 2001.

[4] R. Albert and Barabasi A.-L. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, 2002.

[5] R. Albert and A. L. Barabasi. Topology of evolving networks: local events and universality. *Phys Rev Lett*, 85(24):5234–7, 2000.

[6] R. Albert, H. Jeong, and A. L. Barabasi. Diameter of the world-wide web. *Nature*, 401:387–392, 1999.

[7] R. Albert, H. Jeong, and A.-L. Barabasi. Error and attack tolerance of complex networks. *Nature*, 406:378–382, 2000.

[8] C. J. Alpert and A. B. Kahng. Recent directions in netlist partitioning: a survey. *Integration: the VLSI Journal*, 19:1–81, 1995.

[9] S. F. Altschul, W. Gish, W. Miller, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[10] S. F. Altschul, T. L. Madden, A. A. Schffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.

[11] L. A. N. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley. Classes of behavior of small-world networks. *Proc Natl Acad Sci U S A*, 97:11149–11152, 2000.

[12] M. Ashburner. FlyBase. *Genome News*, 13:19–20, 1993.

[13] G. D. Bader, D. Betel, and C. W. V. Hogue. BIND: the biomolecular interaction network database. *Nucleic Acids Research*, 31(1):248–250, 2003.

[14] G. D. Bader and C. W. V. Hogue. Analyzing yeast protein-protein interaction data obtained from different sources. *Nature Biotechnology*, 20:991–997, 2002.

[15] G. D. Bader and C. W. V. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4:2, 2003.

[16] A. Bairoch and R. Apweiler. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Research*, 28:45–48, 2000.

[17] F. Ball, J. Mollison, and G. Scalio-Tomba. Epidemics with two levels of mixing. *The Annals of Applied Probability*, 7:46–89, 1997.

[18] A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–12, 1999.

[19] A.-L. Barabasi, R. Albert, and H. Jeong. Mean-field theory for scale-free random networks. *Physica A*, 272:173–197, 1999.

[20] A.-L. Barabasi, Z. Dezso, E. Ravasz, Z.-H. Yook, and Z. N. Oltvai. Scale-free and hierarchical structures in complex networks. *Sitges Proceedings on Complex Networks*, 2004. to appear.

[21] A.-L. Barabasi, E. Ravasz, and T. Vicsek. Deterministic scale-free networks. *Physica A*, 299:559–564, 2001.

[22] A. D. Barbour and G. Reinert. Small worlds. *Random Structures and Algorithms*, 19:54–74, 2001.

[23] A. Barrat and M. Weigt. On the properties of small-world network models. *European Physical Journal B*, 13:547–560, 2000.

[24] M. Barthelemy and L. A. N. Amaral. Small-world networks: evidence for crossover picture. *Physical Review Letters*, 82:3180–3183, 1999.

[25] V. Batagelj and A. Mrvar. Pajek – program for large network analysis. *Connections*, 2:47–57, 1998.

[26] A. D. Baxevanis. The molecular biology database collection: 2003 update. *Nucleic Acids Research*, 31(1):1–12, 2003.

[27] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.

[28] E. A. Bender and E. R. Canfield. The asymptotic number of labeled graphs with given degree sequences. *Journal of Combinatorial Theory A*, 24:296–307, 1978.

[29] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, B. A. Rapp, and D. L. Wheeler. GenBank. *Nucleic Acids Research*, 30:17–20, 2002.

[30] M. Blatt, S. Wiseman, and E. Domany. Superparamagnetic clustering of data. *Physical Review Letters*, 76(18):3251–3254, 1996.

[31] B. Bollobas. *Random Graphs*. Academic, London, 1985.

[32] B. Bollobas and O. Riordan. The diameter of a scale-free random graph. *Combinatorica*, 2001. to appear.

[33] M. Boots and A. Sasaki. ’Small worlds’ and the evolution of virulence: infection occurs locally and at a distance. *Proceedings of The Royal Society of London. Series B, Biological Sciences*, 266:1933–1938, 1999.

[34] S. Bornholdt and H. Ebel. World-wide web scaling exponent from simon’s 1955 model. *Physical Review E*, 64:046401, 2001.

[35] B.-J. Breitkreutz, C. Stark, and M. Tyers. The GRID: The general repository for interaction datasets. *Genome Biology*, 4:R23:R23.1–R23.3, 2003.

[36] B.-J. Breitkreutz, C. Stark, and M. Tyers. Osprey: a network visualization system. *Genome Biology*, 4:R22:R22.1–R22.4, 2003.

[37] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure of the web. *Computer Networks*, 33:309–320, 2000.

[38] D. Bu, Y. Zhao, L. Cai, H. Xue, X. Zhu, H. Lu, J. Zhang, S. Sun, L. Ling, N. Zhang, G. Li, and R. Chen. Topological structure analysis of the protein-protein interaction network in budding yeast. *Nucleic Acids Research*, 31(9):2443–2450, 2003.

[39] D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Network robustness and fragility: percolation on random graphs. *Physical Review Letters*, 85:5468–5471, 2000.

[40] C. Chekuri, A. Goldberg, D. Karger, M. Levine, and C. Stein. Experimental study of minimum cut algorithms. In *ACM-SIAM Symposium on Discrete Algorithms (SODA 97)*, pages 324–333, 1997.

[41] S. A. Chervitz. Comparison of the complete protein sets of worm and yeast: orthology and divergence. *Science*, 282:2022–2028, 1998.

[42] F. Chung and L. Lu. The diameter of sparse random graphs. *Advances in Applied Mathematics*, 26:257–279, 2001.

[43] R. Cohen, K. Erez, D. ben Avraham, and S. Havlin. Resilience of the internet to random breakdowns. *Physical Review Letters*, 85:4626–4628, 2000.

[44] The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 2000.

[45] M. C. Costanzo, M. E. Crawford, J. E. Hirschman, J. E. Kranz, P. Olsen, L. S. Robertson, M. S. Skrzypek, B. R. Braun, K. L. Hopkins, P. Kondu, C. Lengieza, J. E. Lew-Smith, M. Tillberg, and J. I. Garrels. YPD, PombelPD and WorkPD: model organism volumes of the BioKnowledge library, and integrated resource for protein information. *Nucleic Acids Research*, 29:75–79, 2001.

[46] T. Dandekar, B. Snel, M. Huynen, and P. Bork. Conservation of gene order: a fingerprint of proteins that physically interact. *Trends Biochem. Sci.*, 23:324–328, 1998.

[47] S. N. Dorogovtsev, A. V. Goltsev, and J. F. F. Mendes. Pseudofractal scale-free web. *Physical Review E*, 65:066122, 2002.

[48] S. N. Dorogovtsev and J. F. F. Mendes. Evolution of networks with aging of sites. *Physical Review E*, 62:1842–1845, 2000.

[49] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. Wiley, New York, 1973.

[50] S. S. Dwight, M. A. Harris, K. Dolinski, C. A. Ball, G. Binkley, K. R. Christie, D. G. Fisk, L. Issel-Tarver, M. Schroeder, G. Sherlok, A. Sethuraman, S. Weng, D. Botstein, and J. M. Cherry. Saccharomyces genome database (SGD) provides secondary gene annotation using the gene ontology (GO). *Nucleic Acids Research*, 30:69–72, 2002.

[51] J. P. Eckmann and E. Moses. Curvature of co-links uncovers hidden thematic layers in the world wide web. *Proc Natl Acad Sci U S A*, 99(9):5825–9, 2002.

[52] A. M. Edwards, B. Kus, R. Jansen, D. Greenbaum, J. Greenblatt, and M. Gerstein. Bridging structural biology and genomics: assessing protein interaction data with known complexes. *Trends Genet*, 18:529–36, 2002.

[53] A. J. Enright. *Computational Analysis of Protein Function within Complete Genomes*. PhD thesis, University of Cambridge, United Kingdom, 2002.

[54] A. J. Enright, I. Iliopoulos, N. C. Kyrpides, and C. A. Ouzounis. Protein interaction maps for complete genomes based on gene fusion events. *Nature*, 402:86–90, 1999.

[55] A. J. Enright, S. Van Dongen, and C. A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, 30(7):1575–1584, 2002.

[56] P. Erdos and A. Renyi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[57] P. Erdos and A. Renyi. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.*, 5:17–61, 1960.

[58] P. Erdos and A. Renyi. On the strength of connectedness of a random graph. *Acta Mathematica Scientia Hungary*, 12:261–267, 1961.

[59] S. Even. *Graph Algorithms*. Computer Science Press, Rockville, Maryland., 1979.

[60] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. *Computer Communications Review*, 29:251–262, 1999.

[61] D. Fell and A. Wagner. The small world of metabolism. *Nature Biotechnology*, 19:1121–1122, 2000.

[62] C. Friedman, P. Kra, H. Yu, M. Krauthammer, and A. Rzhetsky. Genies: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, 17 Suppl. 1:74–82, 2001.

[63] A. C. Gavin, M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J. M. Rick, A. M. Michon, C. M. Cruciat, M. Remor, C. Hofert, M. Schelder, M. Brajenovic, H. Ruffner, A. Merino, K. Klein, M. Hudak, D. Dickson, T. Rudi, V. Gnau, A. Bauch, S. Bastuck, B. Huhse, C. Leutwein, M. A. Heurtier, R. R. Copley, A. Edelmann, E. Querfurth, V. Rybin, G. Drewes, M. Raida, T. Bouwmeester, P. Bork, B. Seraphin, B. Kuster, G. Neubauer, and G. Superti-Furga. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415(6868):141–7, 2002.

[64] H. Ge, Z. Liu, G. M. Church, and M. Vidal. Correlation between transcriptome and interactome mapping data from saccharomyces cerevisiae. *Nat Genet*, 29(4):482–6, 2001.

[65] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link toplogy. *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia*, 1998. ACM Press, New York, NY.

[66] M. Girvan and M. E. Newman. Community structure in social and biological networks. *Proc Natl Acad Sci U S A*, 99(12):7821–6, 2002.

[67] N. Guelzim, S. Bottani, P. Bourgine, and F. Kepes. Topological and causal structure of the yeast transcriptional regulatory network. *Nat Genet*, 31(1):60–3, 2002.

[68] J. Hao and J. Orlin. A faster algorithm for finding the minimum cut in a directed graph. *Journal of Algorithms*, 17(3):424–446, 1994.

[69] E. Hartuv and R. Shamir. An algorithm for clustering cdna fingerprints. *Genomics*, 66(3):249–256, 2000. A preliminary version appeared in Proc. RECOMB '99, pp. 188-197.

[70] E. Hartuv and R. Shamir. A clustering algorithm based on graph connectivity. *Information Processing Letters*, 76(4-6):175–181, 2000.

[71] Y. Ho, A. Gruhler, A. Heilbut, G. D. Bader, L. Moore, S. L. Adams, A. Millar, P. Taylor, K. Bennett, K. Boutilier, L. Yang, C. Wolting, I. Donaldson, S. Schandorff, J. Shewnarane, M. Vo, J. Taggart, M. Goudreault, B. Muskat, C. Alfarano, D. Dewar, Z. Lin, K. Michalickova, A. R. Willems, H. Sassi, P. A. Nielsen, K. J. Rasmussen, J. R. Andersen, L. E. Johansen, L. H. Hansen, H. Jespersen, A. Podtelejnikov, E. Nielsen, J. Crawford, V. Poulsen, B. D. Sorensen, J. Matthiesen, R. C. Hendrickson, F. Gleeson, T. Pawson, M. F. Moran, D. Durocher, M. Mann, C. W. Hogue, D. Figeys, and M. Tyers. Systematic identification of protein complexes in saccharomyces cerevisiae by mass spectrometry. *Nature*, 415(6868):180–3, 2002.

[72] T. Hubbard, D. Barker, E. Birney, G. Cameron, Y. Chen, L. Clark, T. Cox, J. Cuff, V. Curwen, T. Down, R. Durbin, E. Eyras, J. Gilbert, M. Hammond, L. Huminiecki, A. Kasprzyk, H. Lehvaslaiho, P. Lijnzaad, C. Melsopp, E. Mongin, R. Pettett, M. Pocock, S. Potter, A. Rust, E. Schmidt, S. Searle, G. Slater, J. Smith, W. Spooner, A. Stabenau, J. Stalker, E. Stupka, A. Ureta-Vidal, I. Vastrik, and M. Clamp. The ensembl genome database project. *Nucleic Acids Research*, 30:38–41, 2002.

[73] T. R. Hughes, M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett, E. Coffey, H. Dai, Y. D. He, M. J. Kidd, A. M. King, M. R. Meyer, D. Slade, P. Y. Lum, S. B. Stepaniants, D. D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard, and S. H. Friend. Functional discovery via a compendium of expression profiles. *Cell*, 102(1):109–26, 2000.

[74] M. Huynen and P. Bork. Measuring genome evolution. *Proc Natl Acad Sci U S A*, 95:5849–5856, 1998.

[75] R. F. i Chancho and R. V. Sole. The small world of human language. *Proceedings of The Royal Society of London. Series B, Biological Sciences*, 268(1482):2261–2265, 2001.

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

[77] T. Ito, K. Tashiro, S. Muta, R. Ozawa, T. Chiba, M. Nishizawa, K. Yamamoto, S. Kuhara, and Y. Sakaki. Toward a protein-protein interaction map of the budding yeast: A comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proc Natl Acad Sci U S A*, 97(3):1143–7, 2000.

[78] R. Jansen, D. Greenbaum, and M. Gerstein. Relating whole-genome expression data with protein-protein interactions. *Genome Res*, 12(1):37–46, 2002.

[79] H. Jeong, A. L. Barabasi, B. Tombor, and Z. N. Oltvai. The global organization of cellular networks. *submitted*, 2003. http://www.nd.edu/ networks/cell/.

[80] H. Jeong, S. P. Mason, A. L. Barabasi, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–2, 2001.

[81] H. Jeong, Z. N. Oltvai, and A.-L. Barabasi. Prediction of protein essentiality based on genomic data. *ComPlexUs*, 1:19–28, 2003.

[82] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A. L. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–4, 2000.

[83] S. Jones and J. M. Thornton. Protein-protein interactions: a review of protein dimer structures. *Prog. Biophys. Mol. Biol.*, 63:31–65, 1995.

[84] S. Jones and J. M. Thornton. Principles of protein-protein interactions. *Proc Natl Acad Sci U S A*, 93:13–20, 1996.

[85] S. Kauffman. *At Home in the Universe*. Oxford, New York, 1995.

[86] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of Theoretical Biology*, 22:437–467, 1969.

[87] J. J. Keeling. The effects of local spatial structure on epidemiological invasions. *Proceedings of The Royal Society of London. Series B, Biological Sciences*, 266:859–867, 1999.

[88] J. O. Kephart and S. R. White. Directed-graph epidemiological models of computer viruses. *Proc. 1991 IEEE Comput. Soc. Symp. Res. Security Privacy*, pages 343–359, 1991.

[89] J. Kleinberg. Authoritative sources in a hyper-linked environment. *Proceedings of the 9th ACM Conference on Hypertext and Hypermedia*, 1998. ACM Press, New York, NY.

[90] J. M. Kleinberg. Navigation in a small world. *Nature*, 406:845, 2000.

[91] P. L Krapivsky and S. Redner. Organization of growing random networks. *Physical Review E*, 63:066123–1, 2001.

[92] P. L Krapivsky, S. Redner, and F. Leyvraz. Connectivity of growing random networks. *Physical Review Letters*, 85:4629–4632, 2000.

[93] M. Krauthammer, P. Kra, I. Iossifov, S. M. Gomez, G. Hripcsak, V. Hatzivassiloglou, C. Friedman, and A. Rzhetsky. Of truth and pathways: chasing bits of information through myriads of articles. *Bioinformatics*, 18 Suppl. 1:249–257, 2002.

[94] L. F. Largo-Fernandez, R. Huerta, F. Corbancho, and J. Siguenza. Fast response and temporal coherent oscillations in small-world networks. *Physical Review Letters*, 84:2758–2761, 2000.

[95] T. Luczak. Component behavior near the critical point of the random graph process. *Random Structures and Algorithms*, 1:287, 1990.

[96] H. Ma and A.-P. Zheng. Structure and evolution analysis of metabolic networks based on genomic data. *4th Biopathways Consortium Meeting*. Edmonton, Canada, August 1-2, 2002.

[97] M. Mann, R. C. Hendrickson, and A. Pandey. Analysis of proteins and proteomes by mass spectrometry. *Annu. Rev. Biochem.*, 70:437–473, 2001.

[98] S. Maslov and K. Sneppen. Specificity and stability in topology of protein networks. *Science*, 296(5569):910–3, 2002.

[99] D. W. Matula. The cohesive strength of graphs. In G. Chartrand and S. F. Kapoor, editors, *The Many Facets of Graph Theory*, pages 215–221. Lecture Notes in Math., Vol. 110, Springer, Berlin, 1969.

[100] D. W. Matula. Cluster analysis via graph theoretic techniques. In R. C. Mullin, K. B. Reid, and D. P. Roselle, editors, *Proc. Louisiana Conference on Combinatorics, Graph Theory and Computing*, pages 199–212. University of Manitoba, Winnipeg, 1970.

[101] D. W. Matula. k-components, clusters and slicing in graphs. *SIAM J. Applied Math*, 22(3):459–480, 1972.

[102] D. W. Matula. Graph theoretic techniques for cluster analysis algorithms. In J. van Ryzin, editor, *Classification and Clustering*, pages 95–129. Academic Press, New York, 1977.

[103] D. W. Matula. Determining edge connectivity in O(nm) time. In *28th IEEE Symposium on Foundations of Computer Science*, pages 249–251, 1987.

[104] R. M. May. *Stability and Complexity in Model Ecosystems*. Princeton Univ. Press, Princeton, 1973.

[105] P. B. McGarvey, H. Huang, W. C. Barker, B. C. Orcutt, J. S. Garavelli, G. Y. Srinivasarao, L. S. Yeh, C. Xiao, and C. H. Wu. PIR: a new resource for bioinformatics. *Bioinformatics*, 16:290–291, 2000.

[106] D. McShan, S. Rao, and I. Shah. Microbial metabolic pathway inference by heuristic search. *4th Biopathways Consortium Meeting*. Edmonton, Canada, August 1-2, 2002.

[107] K. Mehlhorn and S. Naher. *Leda: A platform for combinatorial and geometric computing*. Cambridge University Press, 1999.

[108] H. W. Mewes, D. Frishman, U. Guldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Munsterkotter, S. Rudd, and B. Weil. MIPS: a database for genomes and protein sequences. *Nucleic Acids Res*, 30(1):31–4, 2002.

[109] S. Milgram. The small world problem. *Psychology Today*, 2:60–67, 1967.

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

[111] M. Molloy and B. Reed. A critical point of random graphs with a given degree sequence. *Random Structures and Algorithms*, 6:161–180, 1995.

[112] M. Molloy and B. Reed. The size of the largest component of a random graph on a fixed degree sequence. *Combinatorics, Probability and Computing*, 7:295–306, 1998.

[113] J. M. Montoya and R. V. Sole. Small world patterns of food webs. *Working paper 00-10-059, Santa Fe Institute*, 2001.

[114] H. Nagamochi and T. Ibaraki. Computing edge connectivity in multigraphs and capacitated graphs. *SIAM J. Discrete Math*, 5:54–66, 1992.

[115] M. E. Newman. Scientific collaboration networks: I. network construction and fundamental results. *Physical Review E*, 64:016131, 2001.

[116] M. E. Newman. Ego-centered networks and the ripple effect. *Social Networks*, 25:83–95, 2003.

[117] M. E. Newman and D. J. Watts. Renormalization group analysis in the small-world network model. *Physics Letters A*, 263:341–346, 1999.

[118] M. E. Newman and D. J. Watts. Scaling and percolation in the small-world network model. *Physical Review E*, 60:7332–7342, 1999.

[119] M. E. J. Newman. Models of the small world: a review. *Journal of Statistical Physics*, 101:819–841, 2000.

[120] M. E. J. Newman. The structure and function of networks. *Computer Physics Communications*, 147:44–45, 2001.

[121] M. E. J. Newman. Random graphs as models of networks. In S. Bornholdt and H. G. Schuster, editors, *Handbook of Graphs and Networks*. Wiley-VHC, Berlin, 2002.

[122] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[123] M. E. J. Newman, C. Mooire, and D. J. Watts. Mean-field solution of the small-world network model. *Physical Review Letters*, 84:3201–3204, 2000.

[124] M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Physical Review E*, 64:026118–1, 2001.

[125] S. Ng and M. Wong. Toward routine automatic pathway discovery from on-line scientific text abstracts. *Genome Informatics*, 10:104–112, 1999.

[126] R. Overbeek, N. Larsen, G. D. Pusch, M. D'Souza, E. Selkov Jr, N. Kyrpides, M Fonstein, N. Maltsev, and E. Selkov. WIT: integrated system for high-throughput genome sequence analysis and metabolic reconstruction. *Nucleic Acids Research*, 28(1):123–125, 2000.

[127] A. Pandey and M. Mann. Proteomics to study genes and genomes. *Nature*, 405:837–846, 2000.

[128] R. Pastor-Satorras and A. Vespignani. Epidemic spreading in scale-free networks. *Physical Review Letters*, 86:3200–3203, 2001.

[129] N. Przulj. Analyzing software call graphs. *Microsoft Research, Redmond, WA*, 2003.

[130] N. Przulj and I. Jurisica. A call graph analysis. *CASCON*, 2003. IBM Toronto Lab, Marknam, Ontario, Canada, October 6-9.

[131] N. Przulj, G. Lee, and I. Jurisica. Functional analysis of large software networks. *IBM Academy: Proactive Problem Prediction, Avoidance and Diagnosis*, 2003. IBM T. J. Watson Research Center, NY.

[132] N. Przulj, D. Wigle, and I. Jurisica. Functional topology in a network of protein interactions. *Bioinformatics*, 2003. to appear.

[133] J.-D. Rain, L. Selig, H. De Reuse, V. Battaglia, C. Reverdy, S. Simon, G. Lenzen, F. Petel, J. Wojcik, V. Schachter, Y. Chemama, A. Labigne, and P. Legrain. The protein-protein interaction map of helicobacter pylori. *Nature*, 409:211–215, 2001.

[134] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A.-L. Barabasi. Hierarchical organization of modularity in metabolic networks. *Science*, 297:1551–5, 2002.

[135] G. Rigaut, A. Shevchenko, B. Rutz, M. Wilm, M. Mann, and B. Seraphin. A generic protein purification method for protein complex characterization and proteome exploration. *Nature Biotechnol.*, 17:1030–1032, 1999.

[136] C. J. Roberts, B. Nelson, M. J. Marton, R. Stoughton, M. R. Meyer, H. A. Bennett, Y. D. He, H. Dai, W. L. Walker, T. R. Hughes, M. Tyers, C. Boone, and S. H. Friend. Signaling and circuitry of multiple mapk pathways revealed by a matrix of global gene expression profiles. *Science*, 287(5454):873–80, 2000.

[137] G. M. Rubin, M. D. Yandell, J. R. Wortman, G. L. G. Miklos, C. R. Nelson, I. K. Hariharan, M. E. Fortini, P. W. Li, R. Apweiler, W. Fleischmann, J. M. Cherry, S. Henikoff, M. P. Skupski, S. Misra, M. Ashburner, E. Birney, M. S. Boguski, T. Brody, P. Brokstein, S. E. Celniker, S. A. Chervitz, D. Coates, A. Cravchik, A. Gabrielian, R. F. Galle, W. M. Gelbart, R. A. George, L. S. B. Goldstein, F. Gong, P. Guan, N. L. Harris, B. A. Hay, R. A. Hoskins, J. Li, Z. Li, R. O. Hynes, S. J. M. Jones, P. M. Kuehl, B. Lemaitre, J. T. Littleton, D. K. Morrison, C. Mungall, P. H. O'Farrell, O. K. Pickeral, C. Shue, L. B. Vosshall, J. Zhang, Q. Zhao, X. H. Zheng, F. Zhong, W. Zhong, R. Gibbs, J. C. Venter, M. D. Adams, and S. Lewis. Comparative genomics of the eukaryotes. *Science*, 287:2204–2215, 2000.

[138] C. H. Schilling, D. Letscher, and B. O. Palsson. Theory for the systematic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *Journal of Theoretical Biology*, 203:229–248, 2000.

[139] B. Schwikowski, P. Uetz, and A. Fields. A network of protein-protein interactions in yeast. *Nature Biotechnology*, 18:1257–1261, 2000.

[140] R. Shamir and R. Sharan. CLICK: A clustering algorithm for gene expression analysis. In S. Miyano, R. Shamir, and T. Takagi, editors, *Currents in Computational Molecular Biology*. Universal Academy Press, 2000.

[141] R. Shamir and R. Sharan. Algorithmic approaches to clustering gene expression data. In T. Jiang, T. Smith, Y. Xu, and M. Zhang, editors, *Current Topics in Computational Biology*. MIT Press, 2001.

[142] R. Sharan and R. Shamir. CLICK: A clustering algorithm with applications to gene expression analysis. In *International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 307–316, 2000.

[143] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics*, 31:64–68, 2002.

[144] H. A. Simon. On a class of skew distribution functions. *Biometrika*, 42:425–440, 1955.

[145] V. Spirin, D. Zhao, and L. A. Mirny. Discovery of protein complexes in the network of protein interactions. In *3rd International Conference on Systems Biology (ICSB)*, 2002. Karolinska Institutet, Stockholm, Sweden, Dec. 13 - 15.

[146] O. Sporns, G. Tononi, and G. M. Edelman. Theoretical neuroanatomy: relating anatomical and functional connectivity in graphs and cortical connection matrices. *Cereb. Cortex*, 10:127–141, 2000.

[147] M. Steffen, A. Petti, J. Aach, P. D'haeseleer, and G. Church. Automated modeling of signal transduction networks. *BMC Bioinformatics*, 2002.

[148] J. Stelling, S. Klamt, K. Bettenbrock, S. Schuster, and E. D. Gilles. Metabolic network structure determines key aspects of functionality and regulation. *Nature*, 420:190–3, 2002.

[149] K. E Stephan. Computational analysis of functional connectivity between areas of primate visual cortex. *Phil. Trans. R. Soc. Lond. B*, 355:111–126, 2000.

[150] M. Stoer and F. Wagner. A simple min-cut algorithm. *Journal of the ACM*, 44(4):585–591, 1997.

[151] G. Stoesser, W. Baker, A. van den Broek, E. Camon, M. Garia-Pastor, C. Kanz, T. Kulikova, R. Leinonen, Q. Lin, V. Lombard, R. Leopez, N. Redaschi, P. Stoehr, M. A. Tuli, K. Tzouvara, and R. Vaughan. The EMBL nucleotide sequence database. *Nucleic Acids Research*, 30:21–26, 2002.

[152] S. H. Strogatz. Exploring complex networks. *Nature*, 410:268–276, 2001.

[153] Y. Tateno, T. Imanishi, S. Miyazaki, K. Fukami-Kobayashi, N. Saitou, H. Sugawara, and T. Gojobori. DAN data bank of japan (DDBJ). *Nucleic Acids Research*, 30:27–30, 2002.

[154] A. H. Tong, B. Drees, G. Nardelli, G. D. Bader, B. Brannetti, L. Castagnoli, M. Evangelista, S. Ferracuti, B. Nelson, S. Paoluzi, M. Quondam, A. Zucconi, C. W. Hogue, S. Fields, C. Boone, and G. Cesareni. A combined experimental and computational strategy to define protein interaction networks for peptide recognition modules. *Science*, 295(5553):321–4, 2002.

[155] Y. Tu. How robust is the internet? *Nature*, 406:353–4, 2000.

[156] Peter Uetz, Loic Giot, Gerard Cagney, Traci A. Mansfield, Richard S. Judson, James R. Knight, Daniel Lockshon, Vaibhav Narayan, Maithreyan Srinivasan, Pascale Pochart, Alia Qureshi-Emili, Ying Li, Brian Godwin, Diana Conover, Theodore Kalbfleish, Govindan Vijayadamodar, Meijia Yang, Mark Johnston, Stanley Fields, and Jonathan M. Rothberg. A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae. *Nature*, 403:623–627, 2000.

[157] S. M. van Dongen. *Graph Clustering by Flow Simulation*. PhD thesis, University of Utrecht, The Netherlands, 2000.

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

[159] A. Wagner and D. Fell. The small world inside large metabolic networks. *Proc. Roy. Soc. London Series B*, 268:1803–1810, 2001.

[160] J. Wallinga, K. J. Edmunds, and M. Kretzschmar. Perspective: human contact patterns and the spread of airborne infectious diseases. *Trends in Microbiology*, 7:372–377, 1999.

[161] T. Walsh. Search in small world. *Proc. 16th Int. Joint Conf. Artif. Intell.*, pages 1172–1177, 1999.

[162] D. J. Watts. *Small Worlds*. Princeton University Press, Princeton, 1999.

[163] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[164] D. B. West. *Introduction to Graph Theory*. Prentice Hall, Upper Saddle River, NJ., 1996.

[165] H. S. Wilf. *Generating Functionology*. Academic, Boston, 1990.

[166] R. J. Williams, E. L. Berlow, J. A. Dunne, A. L. Barabasi, and N. D. Martinez. Two degrees of separation in complex food webs. *Proc Natl Acad Sci U S A*, 99:12913–6, 2002.

[167] I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S. M. Kim, and Eisenberg D. Dip: the database of interacting proteins. *Nucleic Acids Research*, 28(1):289–291, 2000.

[168] I. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S. M. Kim, and Eisenberg D. DIP, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Research*, 30(1):303–305, 2002.

[169] E. Yeger-Lotem and H.. Margalit. Detection of regulatory circuits by integration of protein-protein and protein-dna interaction data. *4th Biopathways Consortium Meeting*. Edmonton, Canada, August 1-2, 2002.

[170] S.-H. Yook, H. Jeong, and A.-L. Barabasi. Modeling the internet's large-scale topology. *Proc Natl Acad Sci U S A*, 99:13382–6, 2002.