CHAPTER 1

CLUSTERING METHODS IN PROTEIN-PROTEIN INTERACTION NETWORK

Chuan Lin, Young-rae Cho, Woo-chang Hwang, Pengjun Pei, Aidong Zhang
Department of Computer Science and Engineering
State University of New York at Buffalo
Email: {chuanlin, ycho8, whwng2, ppei, azhang}@cse.buffalo.edu

With completion of a draft sequence of the human genome, the field of genetics stands on the threshold of significant theoretical and practical advances. Crucial to furthering these investigations is a comprehensive understanding of the expression, function, and regulation of the proteins encoded by an organism. It has been observed that proteins seldom act as single isolated species in the performance of their functions; rather, proteins involved in the same cellular processes often interact with each other. Therefore, the functions of uncharacterized proteins can be predicted through comparison with the interactions of similar known proteins. A detailed examination of the protein-protein interaction (PPI) network can thus yield significant new understanding of protein function. Clustering is the process of grouping data objects into sets (clusters) which demonstrate greater similarity among objects in the same cluster than in different clusters. Clustering in the PPI network context groups together proteins which share a larger number of interactions. The results of this process can illuminate the structure of the PPI network and suggest possible functions for members of the cluster which were previously uncharacterized.

This chapter will begin with a brief introduction of the properties of protein-protein interaction networks, including a review of the data which has been generated by both experimental and computational approaches. A variety of methods which have been employed to cluster these networks will then be presented. These approaches are broadly characterized as either distance-based or

Knowledge Discovery in Bioinformatics: Techniques, Methods and Application . By

graph-based clustering methods. Techniques for validating the results of these approaches will also be discussed.

1.1 INTRODUCTION

1.1.1 Protein-Protein Interaction

1.1.1.1 Proteome in Bioinformatics

With the completion of a draft sequence of the human genome, the field of genetics stands on the threshold of significant theoretical and practical advances. Crucial to furthering these investigations is a comprehensive understanding of the expression, function, and regulation of the proteins encoded by an organism [96]. This understanding is the subject of the discipline of proteomics. Proteomics encompasses a wide range of approaches and applications intended to explicate how complex biological processes occur at a molecular level, how they differ in various cell types, and how they are altered in disease states.

Defined succinctly, proteomics is the systematic study of the many and diverse properties of proteins with the aim of providing detailed descriptions of the structure, function, and control of biological systems in health and disease [68]. The field has burst onto the scientific scene with stunning rapidity over the past several years. Figure 1.1. shows the trend of the number of occurrences of the term “proteome” found in Pubmed bioinformatics citations over the past decade. This figure strikingly illustrates the rapidly-increasing role played by proteomics in bioinformatics research in recent years.

A particular focus of the field of proteomics is the nature and role of interactions between proteins. Protein-protein interactions play diverse roles in biology and differ based on the composition, affinity, and lifetime of the association. Non-covalent contacts between residue sidechains are the basis for protein folding, protein assembly, and protein-protein interaction [65]. These contacts facilitate a variety of interactions and associations within and between proteins. Based on their diverse structural and functional characteristics, protein-protein interactions can be categorized in several ways [64]. On the basis of their interaction surface, they may be homo-oligomeric or hetero-oligomeric; as judged by their stability, they may be obligate or non-obligate; and as measured by their persistence, they may be transient or permanent. A given protein-protein interaction can fall into any combination of these three categorical pairs. An interaction may also require reclassification under certain conditions; for example, it may be mainly transient in vivo but become permanent under certain cellular conditions.

1.1.1.2 Significance of Protein-protein Interaction

It has been observed that proteins seldom act as single isolated species while performing their functions in vivo [91]. The analysis of annotated proteins reveals that proteins involved in the same cellular processes often interact with each other [86]. The function of unknown proteins may be postulated on the basis of their interaction with a known protein target of known function. Mapping protein-protein interactions has not only provided insight into protein function but has facilitated the modeling of functional pathways to elucidate the molecular mechanisms of cellular processes. The study of protein interactions is fundamental to understanding how proteins function within the cell. Characterizing the interactions of proteins in a given cellular proteome will be the next milestone along the road to understanding the biochemistry of the cell.

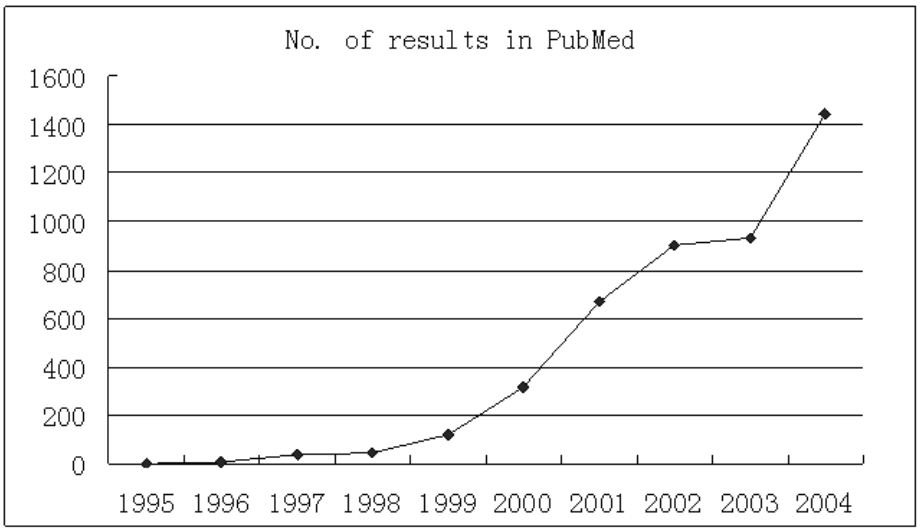


Figure 1.1. Number of results found in PubMed for proteome.

The result of two or more proteins interacting with a specific functional objective can be demonstrated in several different ways. The measurable effects of protein interactions have been outlined by Phizicky and Fields [74]. Protein interactions can:

- alter the kinetic properties of enzymes; this may be the result of subtle changes at the level of substrate binding or at the level of an allosteric effect;
- act as a common mechanism to allow for substrate channeling;
- create a new binding site, typically for small effector molecules;
- inactivate or destroy a protein; or
- change the specificity of a protein for its substrate through interaction with different binding partners; e.g., demonstrate a new function that neither protein can exhibit alone.

Protein-protein interactions are much more widespread than once suspected, and the degree of regulation that they confer is large. To properly understand their significance in the cell, one needs to identify the different interactions, understand the extent to which they take place in the cell, and determine the consequences of the interaction.

1.1.2 Experimental Approaches for PPI Detection

In early reviews, physicochemical approaches for detecting protein-protein interactions included site-directed mutagenesis or chemical modification of amino acid groups participating in such interactions [52, 66, 79, 84]. The following subsections will discuss these bioinformatic and functional proteomic methods. These include predictions of protein-protein interaction via the yeast two-hybrid system, mass spectrometry, and protein microarrays.

1.1.2.1 Yeast Two-hybrid System

One of the most common approaches to the detection of pairs of interacting proteins in vivo is the yeast two-hybrid (Y2H) system [7, 36]. The Y2H system, which was developed by Fields and Song [23], is a molecular-genetic tool which facilitates the study of protein-protein interactions [1]. The interaction of two proteins transcriptionally activates a reporter gene, and a color reaction is seen on specific media. This indication can track the interaction between two proteins, revealing "prey" proteins which interact with a known "bait" protein.

The yeast two-hybrid system enables both highly-sensitive detection of protein-protein interactions and screening of genome libraries to ascertain the interaction partners of certain proteins. The system can also be used to pinpoint protein regions mediating the interactions [37]. However, the classic Y2H system has several limitations. First, it cannot, by definition, detect interactions involving three or more proteins and those depending on post-translational modifications except those applied to the budding yeast itself [37]. Second, since some proteins (for example, membrane proteins) cannot be reconstructed in the nucleus, the yeast two-hybrid system is not suitable for the detection of interactions involving these proteins. Finally, the method does not guarantee that an interaction indicated by Y2H actually takes place physiologically.

Recently, numerous modifications of the Y2H approach have been proposed which characterize protein-protein interaction networks by screening each protein expressed in a eukaryotic cell [24]. Drees [19] has proposed a variant which includes the genetic information of a third protein. Zhang et al. [92] have suggested the use of RNA for the investigation of RNA-protein interactions. Vidal et al. [85] used the URA3 gene instead of GAL4 as the reporter gene; this two-hybrid system can be used to screen for ligand inhibition or to dissociate such complexes. Johnson and Varshavsky [43] have proposed a cytoplasmic two-hybrid system which can be used for screening of membrane protein interactions.

Despite the various limitations of the Y2H system, this approach has revealed a wealth of novel interactions and has helped illuminate the magnitude of the protein interactome. In principle, it can be used in a more comprehensive fashion to examine all possible binary combinations between the proteins encoded by any single genome.

1.1.2.2 Mass Spectrometry Approaches

Another traditional approach to PPI detection is to use quantitative mass spectrometry to analyze the composition of a partially-purified protein complex together with a control purification in which the complex of interest is not enriched.

Mass spectrometry-based protein interaction experiments have three basic components: bait presentation, affinity purification of the complex, and analysis of the bound proteins [2]. Two large-scale studies [25, 35] have been published on the protein-protein interaction network in yeast. Each study attempted to identify all the components that were present in "naturally"-generated protein complexes, which requires essentially pure preparations of each complex [49]. In both approaches, bait proteins were generated that carried a particular affinity tag. In the case studied by Gavin et al. [25], 1,739 TAP-tagged genes were introduced into the yeast genome by homologous recombination. Ho et al. [35] expressed 725 proteins modified to carry the FLAG epitope. In both cases, the proteins were expressed in yeast cells, and complexes were purified using a single immunoaffinity purification step. Both groups resolved the components of each purified complex with a one-dimensional denaturing polyacrylamide gel electrophoresis (PAGE) step. From the 1,167 yeast strains generated by Gavin et al. [25], 589 protein complexes were purified, 232 of which were unique. Ho et al. [35] used 725 protein baits and detected 3,617 interactions that involved 1,578 different proteins.

Mass-spectrometry-based proteomics can be used not only in protein identification and quantification [16, 50, 72, 89] ,but also for protein analysis, which includes protein profiling [51], post-translational modifications (PTMs) [55, 56] and, in particular, identification of protein-protein interactions.

Compared with two-hybrid approaches, mass-spectrometry-based methods are more effective in characterizing highly abundant, stable complexes. MS-based approaches permit the isolation of large protein complexes and the detection of networks of protein interactions. The two-hybrid system is more suited to the characterization of binary interactions, particularly to the detection of weak or transient interactions.

1.1.2.3 Protein Microarray

Microarray-based analysis is a relatively high-throughput technology which allows the simultaneous analysis of thousands of parameters within a single experiment. The key advantage of the microarray format is the use of a nonporous solid surface, such as glass, which permits precise deposition of capturing molecules (probes) in a highly dense and ordered fashion. The early applications of microarrays and detection technologies were largely centered on DNA-based applications. Today, DNA microarray technology is a robust and reliable method for the analysis of gene function [12]. However, gene expression arrays provide no information on protein post-translational modifications (such as phosphorylation or glycosylation) that affect cell function. To examine expression at the protein level and acquire quantitative and qualitative information about proteins of interest, the protein microarray was developed.

A protein microarray is a piece of glass on which various molecules of protein have been affixed at separate locations in an ordered manner, forming a microscopic array [54]. These are used to identify protein-protein interactions, to identify the substrates of protein kinases, or to identify the targets of biologically-active small molecules. The experimental procedure for protein microarray involves choosing solid supports, arraying proteins on the solid supports, and screening for protein-protein interactions.

Experiments with the yeast proteome microarray have revealed a number of protein-protein interactions which had not previously been identified through Y2H or MS-based approaches. Global protein interaction studies were performed with a yeast proteome chip. Ge [26] has described a universal protein array which permits quantitative detection of protein interactions with a range of proteins, nucleic acids, and small molecules. Zhu et al. [95] generated a yeast proteome chip from recombinant protein probes of 5,800 open-reading frames.

1.1.3 Computational Methods to Predict Protein-protein Interaction

The yeast two-hybrid system and other experimental approaches provide a useful tool for the detection of protein-protein interactions occurring in many possible combinations between specified proteins. The widespread application of these methods has generated a substantial bank of information about such interactions. However, the data generated can be erroneous, and these approaches are often not completely inclusive of all possible protein-protein interactions. In order to form an understanding of the total universe of potential interactions, including those not detected by these methods, it is useful to develop an approach to predict possible interactions between proteins. The accurate prediction of protein-protein interactions is therefore an important goal in the field of molecular recognition.

A number of approaches to PPI prediction are based on the use of genome data. Pellegrini et al. [71] introduced the first such method, which predicts an interaction between two

proteins in a given organism if these two proteins have homologs in another organism. A subsequent extension proposed by Marcotte et al. [57, 58] detects co-localization of two genes in different genomes. Two proteins in different organisms are predicted to interact if they have consecutive homologs in a single organism. Dandekar et al. [17] used the adjacency of genes in various bacterial genomes to predict functional relationships between the corresponding proteins. Proteins whose genes are physically close in the genomes of various organisms are predicted to interact.

Jasen et al. [40] investigated the relationship between protein-protein interaction and mRNA expression levels by analyzing existing yeast data from a variety of sources and identifying general trends. Two different approaches were used to analyze the two types of available expression data; normalized differences were computed for absolute expression levels, while a more standard analysis of profile correlations was applied to relative expression levels. This investigation indicated that a strong relationship exists between expression data and most permanent protein complexes.

Some researchers have used data-mining techniques to extract useful information from large data sources. Oyama et al. [67] used a method termed *Association Rules Discovery* to identify patterns and other features from accumulated protein-protein interaction data. This research mined data from four different sources. This aggregated data included 4,307 unique protein interaction pairs. General rules were derived from 5,241 features extracted from the functional, primary-structural, and other aspects of proteins. After transforming the traditional protein-based transaction data into interaction-based transaction data, Oyama was able to detect and articulate 6,367 rules. Of these, 5,271 rules had at least one feature pertaining to sequences. As this potential had been suggested by other researchers, these results confirmed the efficacy of this method.

As mentioned above, experimental and computational approaches have generated significant quantities of PPI data, but these data sets are typically incomplete, contradictory, and include many false positives. It is therefore necessary for improved accuracy to integrate evidence from many different sources for evaluating protein-protein interactions. Jansen et al. [39] proposed a Bayesian approach for integrating interaction information that allows for the probabilistic combination of multiple data sets and demonstrates its application to yeast data. This approach assesses each source for interactions by comparison with samples of known positives and negatives, yielding a statistical measure of reliability. The likelihood of possible interactions for every protein pair is then predicted by combing each independent data source, weighted according to its reliability. The predictions were validated by TAP (tandem affinity purification) tagging experiments. It was observed that, at given levels of sensitivity, the predictions were more accurate than the existing high-throughput experimental data sets.

1.2 PROPERTIES OF PPI NETWORK

Although reductionism has long been the prevailing paradigm guiding the interpretation of experimental results, it has become increasingly evident that a discrete biological function can only rarely be attributed to an individual molecule. Rather, many biological characteristics arise from complex interactions between numerous cellular constituents, such as proteins, DNA, RNA, and small molecules [4, 34, 44, 46]. Therefore, understanding the structure and dynamics of the complex intercellular web of interactions has become a central focus of biological investigation.

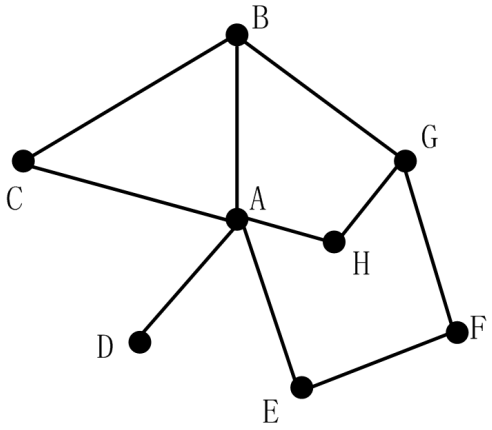


Figure 1.2. A graph in which a node has a degree of 5. Figure is adapted from [9]

1.2.1 PPI Network Representation

An investigation of protein-protein interaction mechanisms begins with the representation and characterization of the PPI network structure. The simplest representation takes the form of a mathematical graph consisting of nodes and edges (or links) [88]. Proteins are represented as nodes in such a graph; two proteins which interact physically are represented as adjacent nodes connected by an edge.

Degree

The **degree** (or connectivity) of a node is the number of other nodes with which it is connected [9]. It is the most elementary characteristic of a node. For example, in the undirected network, Figure 1.2., node A has degree $k = 5$.

Path, shortest path and mean path

The **path** between two nodes is a sequence of adjacent nodes. The number of edges in this path is termed the **path length**, and distances within network are measured in terms of path length. As there are many alternative paths between two nodes, the **shortest path** between the specified nodes refers to the path with the smallest number of links. The **mean path** length of the network represents the average over the shortest paths between all pairs of nodes.

Degree distribution

Graph structures can be described according to numerous characteristics, including the distribution of path lengths, the number of cyclic paths, and various measures to compute clusters of highly-connected nodes [88]. Barabasi and Oltvai [9] introduced the concept of **degree distribution**, $P(k)$, to quantify the probability that a selected node will have exactly k links. $P(k)$ is obtained by tallying the total number of nodes $N(k)$ with k links and dividing this figure by the total number of nodes N. Different network classes can be distinguished by the degree distribution. For example, a random network follows a Poisson distribution. By contrast, a scale-free network has a power-law degree distribution, indicating that a few hubs bind numerous small nodes. Most biological networks are scale-free, with degree distributions approximating a power law, $P(k) \sim k^{-\gamma}$. When $2 \leq \gamma \leq 3$, the

hubs play a significant role in the network [9].

Clustering coefficient

In many networks, if node A is connected to B, and B is connected to C, then A has a high probability of direct linkage to C. Watts [90] quantified this phenomenon using the clustering coefficient, $C_I = 2n_I/k_I(k_I - 1)$, where n_I is the number of links connecting the k_I neighbors of node I to each other. In this coefficient, n_I indicates the number of triangles that pass through node I, and $k_I(k_I - 1)/2$ is the total number of triangles that could pass through node I. For example, in Figure 1.2., $n_A = 1$ and $C_A = 1/10$, while $n_F = 0$, $C_F = 0$.

The average degree, average path length and average clustering coefficient depend on the number of nodes and links in the network. However, the degree distribution $P(k)$ and clustering coefficient $C(k)$ functions are independent of the size of the network and represent its generic features. These functions can therefore be used to classify various network types [9].

1.2.2 Characteristics of Protein-Protein Networks

Scale-free network

Recent publications have indicated that protein-protein interactions have the features of a scale-free network [29, 41, 53, 87], meaning that their degree distribution approximates a power law, $P(k) \sim k^{-\gamma}$. In scale-free networks, most proteins participate in only a few interactions, while a few (termed "hubs") participate in dozens of interactions.

Small-world effect

Protein-protein interaction networks have an characteristic property known as the "small-world effect", which states that any two nodes can be connected via a short path of a few links. The small-world phenomenon was first investigated as a concept in sociology [61] and is a feature of a range of networks arising in nature and technology, including the Internet [3], scientific collaboration networks [63], the English lexicon [77], metabolic networks [22], and protein-protein interaction networks [78, 87]. Although the small-world effect is a property of random networks, the path length in scale-free networks is much shorter than that predicted by the small-world effect [14, 15]. Therefore, scale-free networks are "ultra-small". This short path length indicates that local perturbations in metabolite concentrations could permeate an entire network very quickly.

Disassortativity

In protein-protein interaction networks, highly-connected nodes (hubs) seldom directly link to each other [59]. This differs from the assortative nature of social networks, in which well-connected people tend to have direct connections to each other. By contrast, all biological and technological networks have the property of disassortativity, in which highly-connected nodes are infrequently linked each other.

1.3 CLUSTERING APPROACHES

1.3.1 Significance of Clustering in PPI Network

A cluster is a set of objects which share some common characteristics. Clustering is the process of grouping data objects into sets (clusters) which demonstrate greater similarity

among objects in the same cluster than in different clusters. Clustering differs from classification; in the latter, objects are assigned to predefined classes, while clustering defines the classes themselves. Thus, clustering is an unsupervised classification method, which means that it does not rely on training the data objects in predefined classes.

In protein-protein interaction networks, clusters correspond to two types of modules: protein complexes and functional modules. Protein complexes are groups of proteins that interact with each other at the same time and place, forming a single multi-molecular machine. Functional modules consist of proteins that participate in a particular cellular process while binding to each other at a different time and place.

Clustering in protein-protein interaction networks therefore involves identifying protein complexes and functional modules. This process has the following analytical benefits:

(1) clarification of PPI network structures and their component relationships;

(2) inference of the principal function of each cluster from the functions of its members;

(3) elucidation of possible functions of members in a cluster through comparison with the functions of other members.

1.3.2 Challenges of Clustering on PPI Networks

The classic clustering approaches follow a protocol termed "pattern proximity after feature selection" [38]. Pattern proximity is usually measured by a distance function defined for pairs of patterns. A simple distance measure can often be used to reflect dissimilarity between two patterns, while other similarity measures can be used to characterize the conceptual similarity between patterns. However, in protein-protein interaction networks, proteins are represented as nodes and interactions are represented as edges. The relationship between two proteins is therefore a simple binary value: 1 if they interact, 0 if they do not. This lack of nuance makes it difficult to define the distance between the two proteins. Additionally, a high rate of false positives and the sheer volume of data render problematical to the reliable clustering of PPI networks.

Clustering approaches for PPI networks can be broadly characterized as distance-based or graph-based. Distance-based clustering uses classic clustering techniques and focuses on the definition of the distance between proteins. Graph-based clustering includes approaches which consider the topology of the PPI network. Based on the structure of the network, the density of each subgraph is maximized or the cost of cut-off minimized while separating the graph. This following section will discuss each of these clustering approaches in greater detail.

1.3.3 Distance-based Clustering

1.3.3.1 Distance Measure Based on Coefficient

As discussed in [30], the distance between two nodes (proteins) in a PPI network can be defined as follows. Let X be a set of n elements and let $d_{ij} = d(i, j)$ be a non-negative real function $d : X \times X \rightarrow R^{+}$, which satisfy

(1) $d_{ij} > 0$ for $i \neq j$;

(2) $d_{ij} = 0$ for $i = j$;

(3) $d_{ij} = d_{ji}$ for all i,j where d is a distance measure and $D = \{d_{ij}\}$ is a distance matrix; and

(4) if d_{ij} satisfies triangle inequality $d_{ij} \leq d_{ik} + d_{kj}$, then d is a metric.

In PPI network, the binary vectors $X_i = (x_{i1}, x_{i2}, ..., x_{iN})$ represent the set of protein purifications for N proteins, where x_{ik} is 1 if the i^{th} protein interacts with k^{th} protein (the k^{th} protein is presented in the i^{th} purification) and 0 otherwise. If a distance can be determined which fully accounts for known protein complexes, unsupervised hierarchical clustering methods can be used to accurately assemble protein complexes from the data. Frequently, a distance can be easily obtained from a simple matching coefficient which calculates the similarity between two elements. The similarity value S_{ij} can be normalized between 0 and 1, and the distance can be derived from $d_{ij} = 1 - S_{ij}$. If the similarity value of two elements is high, the spatial distance between them should be short.

Several suitable measures have been proposed for this purpose. These include the Jaccard coefficient [32]:

$$S_{mn} = \frac{X_{mn}}{X_{mm} + X_{nn} - X_{mn}} \tag{1.1}$$

Dice coefficient [32]:

$$S_{mn} = \frac{2X_{mn}}{X_{mm} + X_{nn}} \tag{1.2}$$

Simpson coefficient [32]:

$$S_{mn} = \frac{X_{mn}}{min(X_{mm}, X_{nn})} \tag{1.3}$$

Bader coefficient [8]:

$$S_{mn} = \frac{X_{mn}^2}{X_{mm} \times X_{nn}} \tag{1.4}$$

Maryland Bridge coefficient [62]:

$$S_{mn} = \frac{1}{2}(\frac{X_{mn}}{X_{mm}} + \frac{X_{mn}}{X_{nn}}) \tag{1.5}$$

Korbel coefficient [47]:

$$S_{mn} = \frac{\sqrt{X_{mm}^2 + X_{nn}^2}}{\sqrt{2}X_{mm}X_{nn}}X_{mn} \tag{1.6}$$

Correlation coefficient [20]:

$$S_{mn} = \frac{X_{mn} - n\overline{X}_m\overline{X}_n}{\sqrt{(X_{mm} - n\overline{X}_m{}^2)(X_{nn} - n\overline{X}_n{}^2)}}, \tag{1.7}$$

where $X_{ij} = X_i \bullet X_j$ (dot product of two vectors). The value of S_{mn} ranges from 0 to 1. X_{ij} is equal to the number of bits "on" in both vectors, and X_{ii} is equal to the number of bits "on" in one vector. For example, for the case illustrated in Figure 1.2., the matrix X is:

$$X = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{1.8}$$

To calculate the distance between A and B, d_{12}, $X_{11} = X_1 \bullet X_1 = 5$, $X_{22} = X_2 \bullet X_2 = 3$, $X_{12} = X_1 \bullet X_2 = 1$. The Jaccard coefficient is calculated as: $S_{12} = 1/(5 + 3 - 1) = 0.1429$; the distance is then $d_{12} = 1 - 0.1429 = 0.8571$.

This group of distance-based approaches uses classic distance measurements, which are not quite suitable for high dimensional spaces. In a high dimensional space, the distances between each pair of nodes are almost the same for a large data distribution [10]. Therefore, it is hard to attain ideal clustering results by the simplest distance measurements only.

1.3.3.2 Distance Measure by Network Distance

There are other definitions based on network distance which give more fine-grained distance measurements for these pairs. In the definition given above, the distance value will be 0 for any two proteins not sharing an interaction partner. In [75], each edge in the network is assigned a length of 1. The length of the shortest path (distance) between every pair of vertices in the network is calculated to create an all-pairs-shortest-path distance matrix. Each distance in this matrix is then transformed into an "association", defined as $1/d^2$ where d is the shortest-path distance. This transformation emphasizes local associations (short paths) in the subsequent clustering process. The resulting associations range from 0 to 1. The association of a vertex with itself is defined as 1, while the association of vertices that have no connecting path is defined as 0. Two vertices which are more widely separated in the network will have a longer shortest-path distance and thus a smaller association. The association value can be therefore served as the similarity measure for two proteins.

In [69], authors consider the paths of various lengths between two vertices in a weighted protein interaction network. The weight of an edge reflects its reliability and lies in the range between 0 and 1. The *PathStrength* of a path is defined as the product of the weights of all the edges on the path. Then the *k-length PathStrength between two vertices* is defined as the sum of the PathStrength of all k-length paths between the two vertices. The PathStrength of a path captures the probability that a walk on the path can reach its ending vertex. By summing upon all these paths, the k-length PathStrength between two vertices captures the strength of connections between these two vertices by a k-step walk. Since paths of different lengths should have different impact on the connection between two vertices, the k-length PathStrength is normalized by the k-length maximum possible path stength to get the *k-length PathRatio*. Finally, the *PathRatio* measure between two vertices is defined as the sum of the k-length PathRatios between the two vertices for all $k > 1$. Though this measure is mainly applied in assessing the reliability of detected interactions and predicting potential interactions that are missed by current experiments, it can also be used as a similarity measure for clustering.

Another network distance measure was developed by Zhou [94, 93]. He defined the distance d_{ij} from node i to node j as the average number of steps a Brownian particle takes to reach j from i.

Consider a connected network of N nodes and M edges. Its node set is denoted by $V = \{1, ..., N\}$ and its connection pattern is specified by the generalized adjacency matrix A. If there is no edge between node i and node j, $A_{ij} = 0$; if there is an edge between those nodes, $A_{ij} = A_{ji} > 0$, and its value signifies the interaction strength. The set of nearest neighbors of node I is denoted by E_i. As a Brownian particle moves throughout the network, at each time step it jumps from its present position i to a nearest-neighboring position j. When no additional information about the network is known, the jumping probability $P_{ij} = A_{ij}/\sum_{l=1}^{N} A_{il}$ can be assumed. Matrix P is called the transfer matrix.

The node-node distance d_{ij} from i to j is defined as the average number of steps needed for the Brownian particle to move from i through the network to j. Using simple linear-algebraic calculations, it is obvious that

$$d_{ij} = \sum_{l=1}^{n} (\frac{1}{I - B(j)})_{il}, \tag{1.9}$$

where I is the $N \times N$ identity matrix, and matrix $B(j)$ equals the transfer matrix P, with the exception that $B_{lj}(j) \equiv 0$ for any $l \in V$. The distances from all the nodes in V to node j can thus be obtained by solving the linear algebraic equation

$$[I - B(j)]\{d_{1j}, ..., d_{nj}\}^T = \{1, ..., 1\}^T. \tag{1.10}$$

For example, in the network shown in Figure 1.3., with the set nodes $V = 1, 2, 3, 4$, the adjacency matrix A and transfer matrix P are:

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, P = \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 1/2 & 0 & 1/2 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

$B(j)$ can be derived from P:

$$B(1) = \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 1/2 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, B(2) = \begin{bmatrix} 0 & 0 & 1/3 & 1/3 \\ 1/2 & 0 & 1/2 & 0 \\ 1/2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$
$$B(3) = \begin{bmatrix} 0 & 1/3 & 0 & 1/3 \\ 1/2 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, B(4) = \begin{bmatrix} 0 & 1/3 & 1/3 & 0 \\ 1/2 & 0 & 1/2 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

The distance between any two nodes can be calculated with Equation1.9:

$$D = \{d_{ij}\} = \begin{bmatrix} 8/3 & 2 & 2 & 1 \\ 10/3 & 4 & 8/3 & 13/3 \\ 10/3 & 8/3 & 4 & 13/3 \\ 23/11 & 27/11 & 9/11 & 34/11 \end{bmatrix}.$$

Based on the distance measure, Zhou [93] defined a dissimilarity index to quantify the relationship between any two nearest-neighboring nodes. Nearest-neighboring vertices of the same community tend to have small dissimilarity index, while those belonging to different communities tend to have high dissimilarity index.

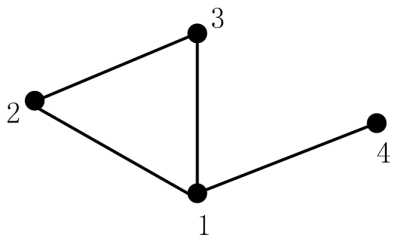


Figure 1.3. Example of distance measure by Brownian particle.

Given two vertices i and j that are nearest neighbors ($A_{ij} > 0$), the difference in their perspectives regarding the network can be quantitatively measured. The dissimilarity index $\Lambda(i, j)$ is defined by the following expression:

$$\Lambda(i,j) = \frac{\sqrt{\sum_{k \neq i,j}^{n} [d_{ik} - d_{jk}]^2}}{n-2}. \tag{1.11}$$

If two nearest-neighboring vertices i and j belong to the same community, then the average distance d_{ik} from i to any another vertex $k(k \neq i, j)$ will be quite similar to the average distance d_{jk} from j to k. This indicates that the perspectives of the network as viewed from i and j will be quite similar. Consequently, $\Lambda(i, j)$ will be small if i and j belong to the same community and large if they belong to different communities.

When this approach is applied to a protein interaction network, clusters of proteins that may be of biological significance can be constructed. Zhou provided three examples of such an application. Most of the proteins in these examples were involved in known functions. It was possible to predict similar biological functions for the few proteins in each cluster which were previously unanalyzed.

1.3.3.3 UVCLUSTER

The UVCLUSTER [6] approach is informed by the observation that the shortest path distance between protein pairs is typically not very fine-grained, and that many pairs have the same distance value. This method proposes an iterative approach to distance exploration; unlike other distance-based approaches, it converts the set of primary distances into secondary distances. The secondary distance measures the strength of the connection between each pair of proteins when the interactions for all the proteins in the group are considered. Secondary distance is derived by first applying a hierarchical clustering step based on the affinity coefficient to generate N different clustering results. The number of solutions generated in which any two selected proteins are not in the same cluster is defined as the secondary distance between the two proteins. Defined succinctly, the secondary distance represents the likelihood that two selected proteins will not be in the same cluster.

This approach has four steps:

1. A **primary distance** d between any two proteins in a PPI network are measured by the minimum number of steps required to connect them. Each valid step is a known, physical protein-protein interaction. Users are allowed to select groups of proteins to be analyzed either by choosing a single protein and establishing a cutoff distance value or by providing the program with a list of proteins.

2. Next, agglomerative hierarchical clustering is applied to the sub-table of primary distances generated in the first step to produce N alternative and equally-valid clustering

solutions. The user specifies a value for N before starting the analysis. UVCLUSTER first randomly samples the elements of the dataset and then clusters them according to the group average linkage. The agglomerative process ends when the affinity coefficient (AC) is reached. The AC is defined as follows:

$$AC = 100[(P_m - C_m)/(P_m - 1)], \tag{1.12}$$

where C_m (the cluster mean) is the average of the distances for all elements included in the clusters and P_m (the partition mean) is the average value of distances for the whole set of selected proteins. AC value is selectly by the user at the start of the process.

3. Once the dataset of N alternative solutions has been obtained, the number of pairs of elements that appear together in the same cluster is counted. A **secondary distance** d' between two elements is defined as the number of solutions in which those two elements do not appear together in the same cluster, divided by the total number of solutions (N). In effect, the secondary distance iteratively resamples the original primary distance data, thus indicating the strength of the connection between two elements. Secondary distance represents the likelihood that each pair of elements will appear in the same cluster when many alternative clustering solutions are generated.

4. After the generation of secondary distance data, the proteins can be clustered using conventional methods such as UPGMA (Unweighted Pair Group Method with Arithmetic Mean) or neighbor-joining. The results of an agglomerative hierarchical clustering process in which UPGMA is applied to the secondary distance data are placed in a second UVCLUSTER output file. A third output file contains a graphical representation of the data in PGM (Portable GreyMap) format. To generate the PGM file, proteins are ordered according to the results described in the second output file.

The use of UVCLUSTER offers four significant benefits. First, the involvement of the secondary distance value facilitates identification of sets of closely-linked proteins. Furthermore, it allows the incorporation of previously-known information in the discovery of proteins involved in a particular process of interest. Third, guided by the AC value, it can establish groups of connected proteins even when some information is currently unavailable. Finally, UVCLUSTER can compare the relative positions of orthologous proteins in two species to determine whether they retain related functions in both of their interactomes.

1.3.3.4 Similarity Learning Method

By incorporating very limited annotation data, a similarity learning method is introduced in [70].

The method defines the similarity between two proteins in a probabilistic framework. Edges in the network are regarded as a means of message passing. Each protein propagates its function to neighboring proteins. Meanwhile, each protein receives these function messages from its neighboring proteins to decide its own function. The final probability of a protein having a specific function is therefore a conditional probability defined on its neighbors' status of having this function annotation. For a certain functional label in consideration, the probability of a protein A having this function is $P(A)$. Another protein B's probability of having this function by propagation using A as the information source can then be represented as a conditional probability $P(B|A)$. This conditional probability gives the capability of A's function being transferred to B via the network. The similarity between proteins A and B is defined as the product of two conditional probabilities:

$$Similarity_{AB} = P(A|B) * P(B|A).$$

Now the problem of estimating the similarity between two proteins is changed into estimating the two conditional probabilities. For this purpose, a statistic model is defined to predict the conditional probabilities using topological features. Since in most organisms, there are certain amount of annotation data for proteins, some training samples are available. The method uses a two-step approach:

1. Model training step: known annotation data are used to estimate the parameters in the model.

2. Conditional probability estimation step: the numerical values of the conditional probabilities are calculated using the model and the parameters estimated in the previous step.

Unsupervised clustering method can be applied on the resulting similarity matrix.

1.3.3.5 Summary

This subsection has provided a review of a series of approaches to distance-based clustering. The first category of approaches uses classic distance measurement methods, which offered a variety of coefficient formulas to compute the distance between proteins in PPI networks. The second class of approaches defines a distance measure based on network distance, including the shortest path length, combined strength of paths of various lengths, and the average number of steps a Brownian particle takes to move from one vertex to another. The third approach type, exemplified by UVCLUSTER, defines a primary and a secondary distance to establish the strength of the connection between two elements in relationship to all the elements in the analyzed dataset. The forth is a similarity learning approach by incorporating some annotation data. Although these four categories of approaches each involve different methods for distance measurement, they all apply classic clustering approaches to the computed distance between proteins.

1.3.4 Graph-based Clustering

A protein-protein interaction network is an unweighted graph in which the weight of each edge between any two proteins is either 1 or 0. This section will explore graph-based clustering, another class of approaches to the process of clustering. Graph-based clustering techniques are explicitly presented in terms of a graph, thus converting the process of clustering a dataset into such graph-theoretical problems as finding a minimum cut or maximal subgraphs in the graph G.

1.3.4.1 Finding Dense Subgraphs

The goal of this class of approaches is to identify the densest subgraphs within a graph; specific methods vary in the means used to assess the density of the subgraphs. Five variations on this theme will be discussed in this subsection.

Enumeration of complete subgraphs

This approach is to identify all fully connected subgraphs (cliques) by complete enumeration [80]. In general, finding all cliques within a graph is an NP-complete problem. Exceptionally, however, this problem is anti-monotonic, meaning that, if a subset of set A

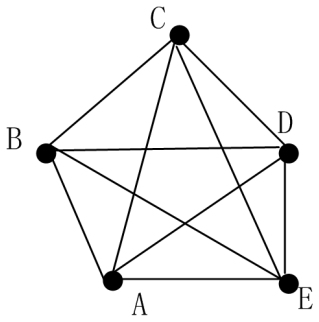


Figure 1.4. Example of Enumeration of complete subgraphs.

is not a clique, then set A is also not a clique. Because of this property, regions of density can be quickly identified in sparse graphs. In fact, to find cliques of size n, one needs only to enumerate those cliques that are of size n-1. Assume a process which starts from the smallest statistically significant number, which is 4 in the case depicted in Figure 1.4. All possible pairs of edges in the nodes will be considered. For example, as shown in Figure 1.4., to examine the edge AB and CD, we must check for edges between AC, AD, BC and BD. If these edges all connect, they are considered fully connected, and a clique $ABCD$ has thus been identified. To test every identified clique $ABCD$, all known proteins will be successively selected. If for protein E, there exist EA, EB, EC, ED, then the clique will be expanded to $ABCDE$. The end result of this process is the generation of cliques which are fully internally connected.

While this approach is simple, it has several drawbacks. The basic assumption underlying the method - that cliques must be fully internally connected - does not accurately reflect the real structure of protein complexes and modules. Dense subgraphs are not necessarily fully connected. In addition, many interactions in the protein network may fail to be detected experimentally, thus leaving no trace in the form of edges.

Monte Carlo optimization

Seeking to address these issues, Spirin and Mirny[80] introduced a new approach which searches for highly-connected rather than fully-connected sets of nodes. This was conceptualized as an optimization problem involving the identification of a set of n nodes that maximizes the object function Q, defined as follows:

$$Q(P) = \frac{2m}{n \cdot (n-1)}. \tag{1.13}$$

The term m enumerates the edges(interactions) among the n nodes in the subgraph P. In this formula, the function Q characterizes the density of a cluster. If the subset is fully connected, Q equals 1; if the subset has no internal edge , Q equals 0. The goal is to find a subset with n nodes which maximizes the objective function Q.

A Monte Carlo approach is used to optimize the procedure. The process starts with a connected subset S of n nodes. These nodes are randomly picked from the graph and then updated by adding or deleting selected nodes from set S, then remain the nodes which increase function Q of S. These steps are repeated until the maximum Q is identified; this yields an n-node subset with high density.

Another quality measure used in this approach is the sum of the shortest distances between selected nodes. Correspondingly, a similar Monte Carlo approach is applied to

minimize this value. This process proceeds as follows. At time $t = 0$, a random set of M nodes is selected. For each pair of nodes i, j from this set, the shortest path L_{ij} between i and j on the graph is calculated. The sum of all shortest paths L_{ij} from this set is denoted as L_0. At each time step, one of M nodes is randomly selected and replaced by random one from among its neighbors. To assess whether the original node is to be replaced by this neighbor, the new sum of all shortest paths, L_1, is then calculated. If $L_1 < L_0$, the replacement is accepted with probability 1. If $L_1 > L_0$, the replacement is accepted with probability $exp^{-\frac{L_1 - L_0}{T}}$, where T is the effective temperature. At every tenth time step, an attempt is made to replace one of the nodes from the current set with a node that has no edges with the current set. This procedure ensures that the process is not caught in an isolated disconnected subgraph. This process is repeated either until the original set converges to a complete subgraph or for a predetermined number of steps. The tightest subgraph, defined as the subgraph corresponding to the smallest L_0, is then recorded. The recorded clusters are merged and redundant clusters are removed. The use of a Monte Carlo approach allows smaller pieces of the cluster to be separately identified rather focusing exclusively on the whole cluster. Monte Carlo simulations are therefore well suited to recognizing highly dispersed cliques.

The experiments conducted by Spirin started with the enumeration of all cliques of size 3 and larger in a graph with $N = 3,992$ nodes and $M = 6,500$ edges. Additionally, 1,000 random graphs of the same size and degree distribution were constructed for comparison. Using the approach described above, more than 50 protein clusters of sizes from 4 to 35 were identified. In contrast, the random networks contained very few such clusters. This work indicated that real complexes have many more interactions than the tightest complexes found in randomly-rewired graphs. In particular, clusters in a protein network have many more interactions than their counterparts in random graphs.

Redundancies in PPI network

Samanta and Liang [76] took a statistical approach to the clustering of proteins. This approach assumes that two proteins that share a significantly larger number of common neighbors than would arise randomly will have close functional associations. This method first ranks the statistical significance of forming shared partnerships for all protein pairs in the PPI network and then combines the pair of proteins with least significance. The p-value is used to rank the statistical significance of the relationship between two proteins. In the next step, the two proteins with smallest p-value are combined and are considered to be in the same cluster. This process is repeated until a threshold is reached. The steps of the algorithm are described in more detail as follows:

First, the p-values [81] for all possible protein pairs are computed and stored in a matrix. The formula of computing p-value between two proteins is shown in Equation 1.14:

$$\begin{aligned} P(N, n_1, n_2, m) &= \frac{\binom{N}{m}\binom{N-m}{n_1-m}\binom{N-n_1}{n_2-m}}{\binom{N}{n_1}\binom{N}{n_2}} \\ &= \frac{\binom{n_1}{m}\binom{N-n_1}{n_2-m}}{\binom{N}{n_2}} \\ &= \frac{(N-n_1)!\,(N-n_2)!\,n_1!\,n_2!}{N!\,m!\,(n_1-m)!\,(n_2-m)!\,(N-n_1-n_2+m)!}, \end{aligned} \tag{1.14}$$

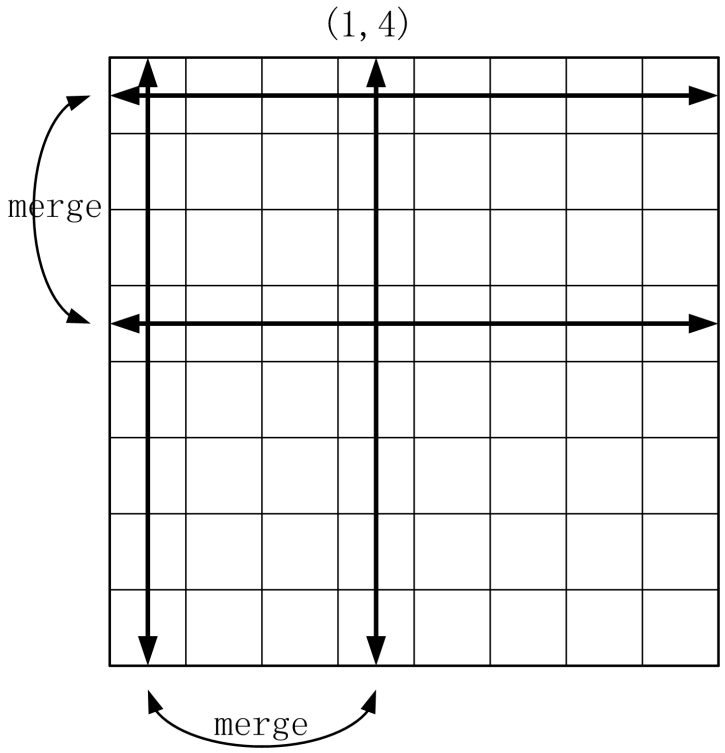


Figure 1.5. If the element (m, n) has the lowest p-value, a cluster is formed with proteins m and n. Therefore, rows/columns m and n are merged with new p-value of the merged row/column as geometric mean of the separate p-values of the corresponding elements. Figure is adapted from [76]

where N is the number of the proteins in the network, each protein in the pair has n_1 and n_2 neighbors, respectively, and m is the number of neighbors shared by both proteins. This formula is symmetric with respect to interchange of n_1 and n_2. It is a ratio in which the denominator is the total number of ways that two proteins can have n_1 and n_2 neighbors. In the numerator, the first term represents the number of ways by which m common neighbors can be chosen from all N proteins. The second term represents the number of ways by which $n_1 - m$ remaining neighbors can be selected from the remaining $N - m$ proteins. The last term represents the number of ways by which $n_2 - m$ remaining neighbors can be selected, none of which can match any of the n_1 neighbors of the first protein.

Second, the protein pair with the lowest p-value is designated as the first group in the cluster. As illustrated in Figure 1.5., the rows and columns for these two proteins are merged into one row and one column. The probability values for this new group are the geometric means of the two original probabilities (or the arithmetic means of the $logP$ values). This process is repeated until a threshold is reached, adding elements to increase the size of the original cluster. The protein pair with the second-lowest p-value is selected to generate the next cluster.

As mentioned in Section 3.2, a high rate of false positives typically creates significant noise which disrupts the clustering of protein complexes and functional modules. This method overcomes this difficulty by using a statistical technique that forms reliable functional associations between proteins from noisy interaction data. The statistical significance

of forming shared partnerships for all protein pairs in the interaction network is ranked. This approach is grounded on the hypothesis that two proteins with a significantly larger number of common interaction pairs in the measured dataset than would arise randomly will also have close functional links[76] .

To validate this hypothesis, all possible protein pairs were ranked in the order of their probabilities. For comparison, the corresponding probabilities were examined for a random network with the same number of nodes and edges but with different connections. The connections in the random network were generated from a uniform distribution. The comparison suggests that the associations in the real dataset contain biologically meaningful information. It also indicates that such low-probability associations did not arise simply from the scale-free nature of the network.

Molecular complex detection (MCODE)

Molecular complex detection (MCODE), proposed by Bader and Hogue [8], is an effective approach for detecting densely-connected regions in large protein-protein interaction networks. This method weights a vertex by local neighborhood density, chooses a few seeds with high weight, and isolates the dense regions according to given parameters. The MCODE algorithm operates in three steps: vertex weighting, complex prediction, and optional postprocessing to filter or add proteins to the resulting complexes according to certain connectivity criteria.

In the first step, all vertices are weighted based on their local network density using the highest k-core of the vertex neighborhood. The core-clustering coefficient of a vertex v is defined to be the density of the highest k-core of the vertices connected directly to v and also v itself (which is called immediate neighborhood of v). Compared with the traditional clustering coefficient, the core-clustering coefficient amplifies the weighting of heavily-interconnected graph regions while removing the many less-connected vertices that are usually part of a biomolecular interaction network. For each vertex v, the weight of v is:

$$w = k \times d, \tag{1.15}$$

where d is the density of the highest k-core graph from the set of vertices including all the vertices directly connected with v and vertex v itself. For example, using the example provided in Figure 1.2., the 2-core weight of node A is $2 \times \frac{2\times 5}{5\times(5-1)} = 1$. It should be noted that node D is not included in the 2-core node set because the degree of node D is 1.

The second step of the algorithm is the prediction of molecular complexes. With a vertex-weighted graph as input, a complex with the highest-weighted vertex is selected as the seed. Once a vertex is included, its neighbors are recursively inspected to determine if they are part of the complex. Then the seed is expanded to a complex until a threshold is encountered. The algorithm assumes that complexes cannot overlap (this condition is more fully addressed in step three), so a vertex is not checked more than once. This process stops when, as governed by the specified threshold, no additional vertices can be added to the complex. The vertices included in the complex are marked as having been examined. This process is repeated for the next-highest unexamined weighted vertex in the network. In this manner, the densest regions of the network are identified. The vertex weight threshold parameter defines the density of the resulting complex.

Post-processing occurs optionally in the third step of the algorithm. Complexes are filtered out if they do not contain at least a 2-core node set. The algorithm may be run with the "fluff" option, which increases the size of the complex according to a given fluff parameter between 0.0 and 1.0. For every vertex v in the complex, its neighbors are added to

the complex if they have not yet been examined and if the neighborhood density (including v) is higher than the given fluff parameter. Vertices that are added by the fluff parameter are not marked as examined, so there can be overlap among predicted complexes with the fluff parameter set.

Evaluated using the Gavin [25] and MIPS [60] data set, MCODE effectively finds densely-connected regions of a molecular interaction network based solely on connectivity data. Many of these regions correspond to known molecular complexes.

Summary

This subsection has introduced a series of graph-based clustering approaches which are structured to maximize the density of subgraphs. The first approach examined seeks to identify fully-connected subgraphs within the network. The second approach improves upon this method by optimizing a density function for finding highly-connected rather than fully-connected subgraphs. The third approach merges pairs of proteins with the lowest p-values, indicating that those proteins have a strong relationship, to identify the dense subgraphs within the network. The final approach discussed assigns each vertex a weight to represent its density in the entire graph and uses the vertex with the highest weight as the seed to generate to a dense subgraph. These approaches all use the topology of the graph to find a dense subgraph within the network and to maximize the density of each subgraph.

1.3.4.2 Finding Minimum Cut

A second category of graph-based clustering approaches generates clusters by trimming or cutting a series of edges to divide the graph into several unconnected subgraphs. Any edge which is removed should be the least important (minimum) in the graph, thus minimizing the informational cost of removing the edges. Here, the least important is based on the structure of the graph. It doesn't mean the interaction between these two proteins is not important. This subsection will present several techniques which are based upon this method.

Highly connected subgraph (HCS) algorithm

The Highly-connected subgraph or HCS method [33] is a graph-theoretic algorithm which separates a graph into several subgraphs using minimum cuts. The resulting subgraphs satisfy a specified density threshold. Despite its interest in density, this method differs from approaches discussed earlier which seek to identify the densest subgraphs. Rather, it exploits the inherent connectivity of the graph and cuts the most unimportant edges to find highly-connected subgraphs.

Some graph-theoretic concepts should first be defined at this point. The $edge\text{-}connectivity$ $k(G)$ of a graph G is the minimum number k of edges whose removal results in a disconnected graph. If $k(G) = l$ then G is termed an l-connected or l-connectivity graph. For example, in Figure 1.6., the graph G is a 2-connectivity graph because we need at least cut two edges (dashed lines in graph) to produce a disconnected graph. A $highly\,connected\,subgraph$ (HCS) is defined as a subgraph whose $edge\text{-}connectivity$ exceeds half the number of vertices. For example, in Figure 1.6., graph G_1 is a $highly\ connected\ subgraph$ because its $edge\text{-}connectivity\ k(G) = 3$ is more than half of the vertices number. A cut in a graph is a set of edges whose removal disconnects the graph. A $minimumcut$ (abbreviated $mincut$) is a cut with a minimum number of edges. Thus, a cut S is a minimum cut of a non-trivial graph $G\ iff\ |S| = k(G)$. The length of a path between two vertices consists of the number of edges in the path. The distance $d(u, v)$ between vertices u and v in graph G is the minimum length of their connecting path, if such path exists; otherwise $d(u, v) = \infty$. The diameter of a connected graph G, denoted $diam(G)$, is the longest distance between any two vertices in G. The degree of vertex v in a graph, denoted $deg(v)$, is the number of edges incident to the vertex.

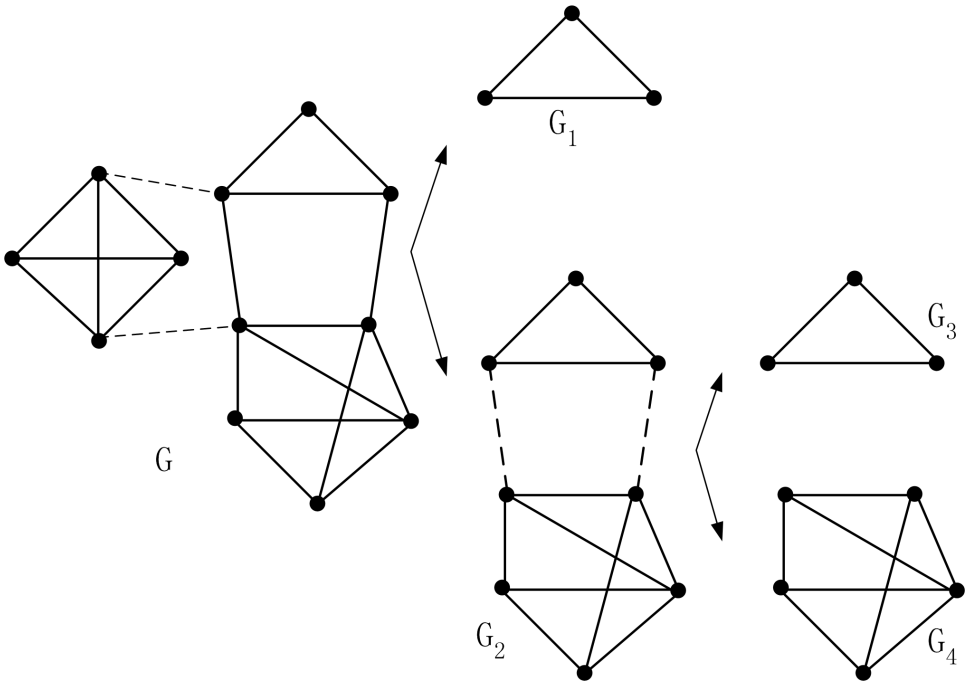


Figure 1.6. An example of applying the HCS algorithm to a graph. Minimum cut edges are denoted by broken lines. Figure is adapted from [33]

The algorithm identifies highly-connected subgraphs as clusters. The HCS algorithm is detailed below, and Figure 1.6. contains an example of its application. Graph G is first separated into two subgraphs G_1 and G_2, which G_1 is a $highly\ connected\ subgraph$ and G_2 is not. Subgraph G_2 is separated into subgraphs G_3 and G_4. This process produces three $highly\ connected\ subgraphs$ G_1, G_3 and G_4, which are considered clusters.

HCS$(G(V,E))$ **algorithm**
```
begin
    (H, H̄, C) ← MINCUT (G)
    if G is highly connected
        then return(G)
    else
        HCS(H)
        HCS(H̄)
end
```

The HCS algorithm generates solutions with desirable properties for clustering. The algorithm has low polynomial complexity and is efficient in practice. Heuristic improvements made to the initial formulation have allowed this method to generate useful solutions for problems with thousands of elements in a reasonable computing time.

Restricted Neighborhood Search Clustering Algorithm (RNSC)

In [45], King et al. proposed a cost-based local search algorithm based on the tabu search metaheuristic [31]. In the algorithm, a clustering of a graph $G = (V, E)$ is defined as a partitioning of the node set V. The process begins with an initial random or user-input clustering and defines a cost function. Nodes are then randomly added to or removed from

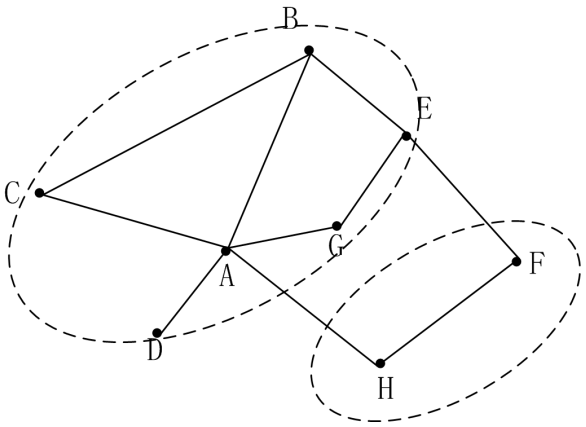


Figure 1.7. An example of RNSC approach.

clusters to find a partition with minimum cost. The cost function is based on the number of invalid connections. An invalid connection incident with v is a connection that exists between v and a node in a different cluster, or, alternatively, a connection that does not exist between v and a node u in the same cluster as v.

The process begins with an initial random or user-input clustering and defines a cost function. Nodes are then randomly added to or removed from clusters to find a partition with minimum cost. The cost function is based on the number of invalid connections.

Consider a node v in a graph G, and a clustering C of the graph. Let α_v be the number of invalid connections incident with v. The naive cost function of C is then defined as

$$C_n(G, C) = \frac{1}{2} \sum_{v \in V} \alpha_v, \tag{1.16}$$

where V is the set of nodes in G. For a vertex v in G with a clustering C, let β_v be the size of the following set: v itself, any node connected to v, and any node in the same cluster as v. This measure reflects the size of the area that v influences in the clustering. The scaled cost function of C is defined as:

$$C_n(G, C) = \frac{|V| - 1}{3} \sum_{v \in V} \frac{\alpha_v}{\beta_v}. \tag{1.17}$$

For example, in Figure 1.7., if the eight vertices are grouped into two clusters as shown, the naive cost function $C_n(G, C) = 2$, and the scaled cost function $C_n(G, C) = \frac{20}{9}$.

Both cost functions seek to define a clustering scenario in which the nodes in a cluster are all connected to one another and there are no other connections between two clusters. The RNSC approach searches for a low-cost clustering solution by optimizing an initial state. Starting with an initial clustering defined randomly or by user input, the method iteratively moves a node from one cluster to another in a random manner. Since the RNSC is randomized, different runs on the same input data will result in different clustering results. To achieve high accuracy in predicting true protein complexes, the RNSC output is filtered according to a maximum P-value selected for functional homogeneity, a minimum density

value, and a minimum size. Only clusters that satisfy these three criteria are presented as predicted protein complexes.

Super paramagnetic clustering (SPC)

The super-paramagnetic clustering (SPC) method uses an analogy to the physical properties of an inhomogenous ferromagnetic model to find tightly-connected clusters in a large graph [11, 27, 28]. Every node on the graph is assigned a Potts spin variable $S_i = 1, 2, ..., q$. The value of this spin variable Si engages in thermal fluctuations which are determined by the temperature T and the spin values of the neighboring nodes. Two nodes connected by an edge are likely to have the same spin value. Therefore, the spin value of each node tends to align itself with that of the majority of its neighbors.

The SPC procedure proceeds via the following steps:

1. Assign to each point $\overrightarrow{x_i}$ a q-state Potts spin variable S_i.

2. Find the nearest neighbors of each point according to a selected criterion; measure the average nearest-neighbor distance a.

3. Calculate the strength of the nearest-neighbor interactions using Eq. (19).

$$J_{ij} = J_{ji} = \frac{1}{\hat{K}} exp(-\frac{\|\overrightarrow{x_i} - \overrightarrow{x_j}\|^2}{2a^2}), \tag{1.18}$$

   where $\hat{K}$ is the average number of neighbors per site.

4. Use an efficient Monte Carlo procedure with Eq. (20) to calculate the susceptibility χ.

$$\chi = \frac{N}{T}(\langle m^2 \rangle - \langle m \rangle^2), m = \frac{(N_{max}/N)q - 1}{q - 1}, \tag{1.19}$$

   where $N_{max} = max\{N_1, N_2, , N_q\}$ and N_μ is the number of spins with the value μ.

5. Identify the range of temperatures corresponding to the superparamagnetic phase, between T_{fs}, the temperature of maximal χ, and the (higher) temperature T_{ps} where χ diminishes abruptly. Cluster assignment is performed at $T_{clus} = (T_{fs} + T_{ps})/2$.

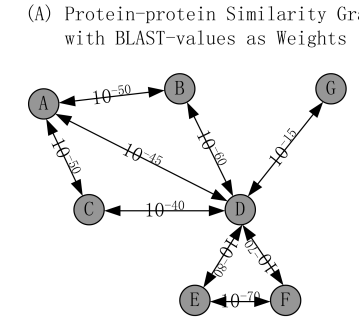
6. Once the J_{ij} have been determined, the spin-spin correlation function can be obtained by a Monte Carlo procedure. Measure at $T = T_{clus}$ the spin-spin correlation function, $\langle \delta_{S_i,S_j} \rangle$, for all pairs of neighboring points $\overrightarrow{x_i}$ and $\overrightarrow{x_j}$.

7. Clusters are identified according to a thresholding procedure. If $\langle \delta_{S_i,S_j} \rangle > \theta$, points $\overrightarrow{x_i}$, $\overrightarrow{x_j}$, are defined as "friends". Then all mutual friends (including friends of friends, etc.) are assigned to the same cluster.

The SPC algorithm is robust in conditions with noise and initialization errors and has been shown to identify natural and stable clusters with no requirement for pre-specifying the number of clusters. Additionally, clusters of any shape can be identified.

Markov clustering

The Markov clustering (MCL) algorithm was designed specifically for application to simple and weighted graphs [82] and was initially used in the field of computational graph clustering [83]. The MCL algorithm finds cluster structures in graphs by a mathematical bootstrapping procedure. The MCL algorithm simulates random walks within a graph by



Generate weighted transition matrix

	A	B	C	D	E	F	G
A	100	50	50	45	0	0	0
B	50	100	0	60	0	0	0
C	50	0	100	40	0	0	0
D	45	60	40	100	80	70	15
E	0	0	0	80	100	70	0
F	0	0	0	70	70	100	0
G	0	0	0	15	0	0	100

(B) Weighted Transition Matrix

Transform weights into column-wise transition probablities

	A	B	C	D	E	F	G
A	0.42	0.24	0.20	0.11	0.00	0.00	0.00
B	0.20	0.48	0.24	0.15	0.00	0.00	0.00
C	0.20	0.00	0.40	0.10	0.00	0.00	0.00
D	0.18	0.28	0.16	0.24	0.32	0.29	0.13
E	0.00	0.00	0.00	0.19	0.40	0.29	0.00
F	0.00	0.00	0.00	0.17	0.28	0.42	0.00
G	0.00	0.00	0.00	0.04	0.00	0.00	0.87

(C) Markov Matrix

Figure 1.8. (A) Example of a protein-protein similarity graph for seven proteins (A-F), circles represent proteins (nodes) and lines (edges) represent detected BLASTp similarities with E-values (also shown). (B) Weighted transition matrix for the seven proteins shown in (A). (C)Associated column stochastic Markov matrix for the seven proteins shown in (A). Figure is adapted from [21]

the alternation of expansion and inflation operations. Expansion refers to taking the power of a stochastic matrix using the normal matrix product. Inflation corresponds to taking the Hadamard power of a matrix (taking powers entrywise), followed by a scaling step, so that the resulting matrix is again stochastic.

Enright et al. [21] employed the MCL algorithm for the assignment of proteins to families. A protein-protein similarity graph is represented as described in Section 2 and as illustrated in Figure 1.8.A. Nodes in the graph represent proteins that are desirable clustering candidates, while edges within the graph are weighted according to a sequence similarity score obtained from an algorithm such as BLAST [5]. Therefore, the edges represent the degree of similarity between these proteins.

A Markov matrix (as shown in Figure 1.8.B) is then constructed in which each entry in the matrix represents a similarity value between two proteins. Diagonal elements are set arbitrarily to a “neutral” value and each column is normalized to produce a column total of 1. This Markov matrix is then provided as input to the MCL algorithm.

As noted above, the MCL algorithm simulates random walks within a graph by alternating two operators: expansion and inflation. The structure of the MCL algorithm is described by the flowchart in Figure 1.9.. After parsing and normalization of the similarity matrix,

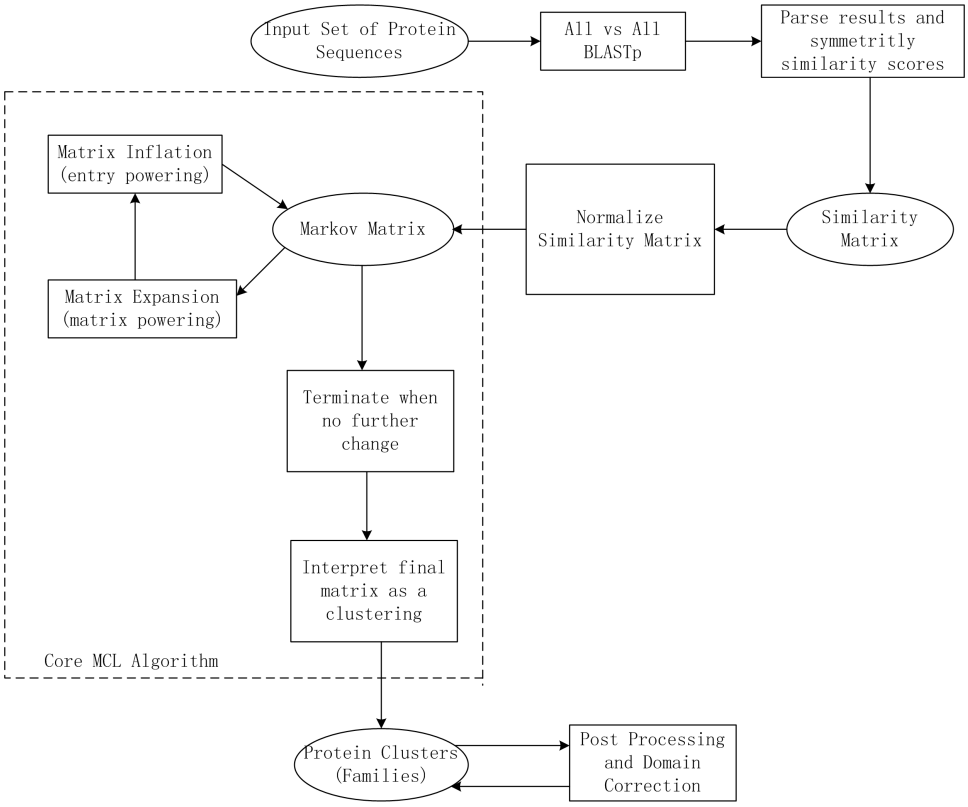


Figure 1.9. This flowchart of the TRIBE-MCL algorithm is from [21] with permission from Oxford University Press.

the algorithm starts by computing the graph of random walks of an input graph, yielding a stochastic matrix. It then uses iterative rounds of the expansion operator, which takes the squared power of the matrix, and the inflation operator, which raises each matrix entry to a given power and then rescales the matrix to return it to a stochastic state. This process continues until there is no further change in the matrix. At last, the final matrix is interpreted as protein clusters with some post processing and domain correction.

Given a matrix $M \in R^{k\times k}, M > 0$, and a real number, $r > 1$, the column stochastic matrix resulting from inflating each of the columns of M with power coefficient r is denoted by $\Gamma_r M$, and Γ_r represents the inflation operator with power coefficient r. Formally the action of $\Gamma_r : R^{k\times k} \to R^{k\times k}$ is defined by:

$$(\Gamma_r M)_{pq} = (M_{pq})^r / \sum_{i=1}^{k} (M_{iq})^r \tag{1.20}$$

Each column j of a stochastic matrix M corresponds with node j of the stochastic graph associated with the probability of moving from node j to node i. For values of $r > 1$, inflation changes the probabilities associated with the collection of random walks departing from one particular node by favoring more probable over less probable walks.

Here expansion and inflation are iteratively used in the MCL algorithm to strengthen the graph where it is strong and to weaken where it is weak until equilibrium is reached. At this point, clusters can be identified according to a threshold. If the weight between two proteins is less than the threshold, the edge between them can be deleted. An important advantage of the algorithm is its "bootstrapping" nature, retrieving cluster structure via the imprint made by this structure on the flow process. Additionally, the algorithm is fast and very scalable, and its accuracy is not compromised by edges between different clusters. The mathematics underlying the algorithm is indicative of an intrinsic relationship between the process it simulates and cluster structure in the input graph.

Line graph generation

Pereira-Leal et al. [73] expressed the network of proteins (e.g., nodes) connected by interactions (e.g., edges) as a network of connected interactions. Figure 1.10.(a) exemplifies an original protein interaction network graph, in which the nodes represent proteins and the edges represent interactions. Periera-Leal's method generates from this an associated line graph, such as that depicted in Figure 1.10.(b), in which edges now represent proteins and nodes represent interactions. This simple procedure is commonly used in graph theory.

First, the protein interaction network is transformed into a weighted network, where the weights attributed to each interaction reflect the degree of confidence attributed to that interaction. Confidence levels are determined by the number of experiments as well as the number of different experimental methodologies that support the interaction. Next, the network connected by interactions is expressed as a network of interactions, which is known in graph theory as a line graph. Each interaction is condensed into a node that includes the two interacting proteins. These nodes are then linked by shared protein content. The scores for the original constituent interactions are then averaged and assigned to each edge. Finally, an algorithm for clustering by graph flow simulation, TribeMCL [21], is used to cluster the interaction network and then to reconvert the identified clusters from an interaction-interaction graph back to a protein-protein graph for subsequent validation and analysis.

This approach focuses on structure of the graph itself and what it represents. It has been included here among the graph-based minimum cutting approaches because it employs the MCL method for clustering. This approach has a number of attractive features. It does not sacrifice informational content, because the original bidirectional network can be recovered at the end of the process. Furthermore, it takes into account the higher-order local neighborhood of interactions. Additionally, the graph it generates is more highly structured than the original graph. Finally, it produces an overlapping graph partitioning of the interaction network, implying that proteins may be present in multiple functional modules. Many other clustering approaches cannot place elements in multiple clusters. This represents a significant inability on the part of those approaches to represent the complexity of biological systems, where proteins may participate in multiple cellular processes and pathways.

Pereira-Leal's group used the protein interaction network derived from the yeast subset of the Database of Interacting Proteins (DIP), which consists of 8,046 physical interactions involving 4,081 yeast proteins. For each protein in a cluster, the research team obtained manually-derived regulatory and metabolic classifications (KEGG), automatic functional classifications (GQFC), and cellular localization information (LOC) from KEGG, GeneQuiz, and MIPS, respectively. On average, the coverage of clusters is 20 regulatory and metabolic roles in KEGG, 45 classes in GeneQuiz, and 48 MIPS.

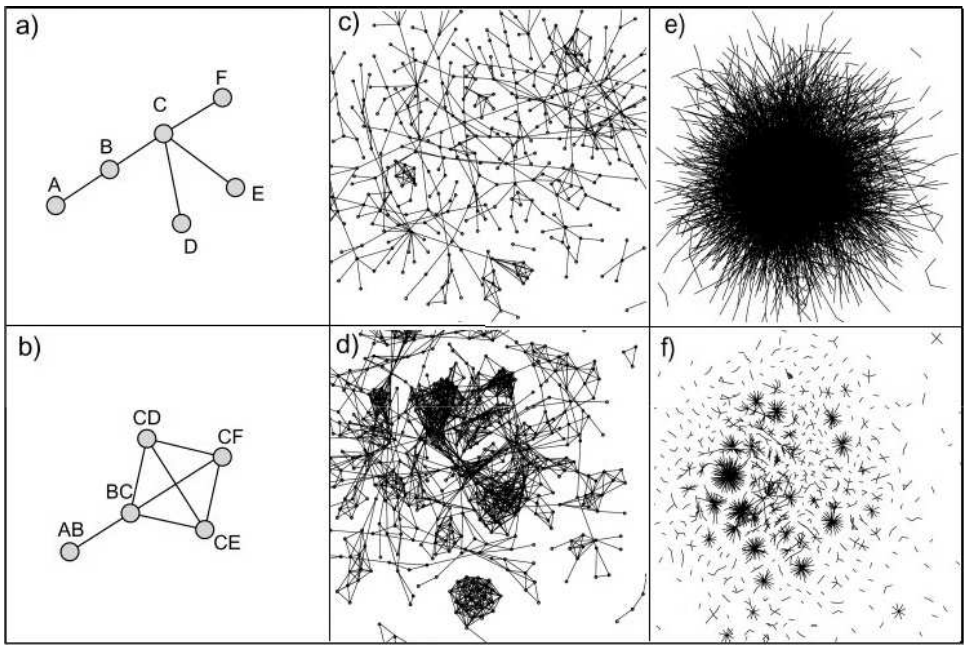


Figure 1.10. Transforming a network of proteins to a network of interactions. (a) Schematic representation illustrating a graph representation of protein interactions: nodes correspond to proteins and edges to interactions. (b) Schematic representation illustrating the transformation of the protein graph connected by interactions to an interaction graph connected by proteins. Each node represents a binary interaction and edges represent shared proteins. Note that labels that are not shared correspond to terminal nodes in (a) in this particular case, A, D, E, and F in edges AB, CD, CE, CF. (c) Graph illustrating a section of a protein network connected by interactions. (d) Graph illustrating the increase in structure as an effect of transforming the protein graph in (c) to an interaction graph. (e) Graph representation of Yeast protein interactions in DIP. (f) Graph representing a pruned version of (e) with the reconstituted interactions after transformation and clustering, as descr ibed in Materials and Methods. These graphs were produced by using BioLayout. Figure is from [73] with permission from Wiley-Liss, Inc., A Wiley Company.

Summary

This subsection has profiled a selection of graph-based clustering approaches which minimize the cost of cutting edges. The first approach discussed defines a highly-connected subgraph and then repeatedly performs a minimum cut until all subgraphs are highly connected. The second approach efficiently searches the space of partitions of all nodes and assigns each a cost function related to cutting the edges in the graph. Identification of the lowest-cost partitions becomes synonymous with finding those clusters with minimum cutting. The third approach assigns each node a Potts spin value and computes the spin-spin correlation function. If the correlation between two spins exceeds a threshold, the two proteins are assigned to the same cluster. The MCL algorithm, which was the fourth presented approach, uses iterative rounds of expansion and inflation to promote flow through the graph where it is strong and to remove flow where it is weak. Clusters are then generated via minimum cutting. The final approach discussed transforms the network of proteins connected by interactions into a network of connected interactions and then uses the MCL algorithm to cluster the interaction network. The first two approaches use the topology of the network to remove the edges in the network; in these methods, the edges have no weight. The other approaches assign each edge a weight which represents the similarity of two proteins; edges with low weights are then cut.

1.4 VALIDATION

So far, this chapter has reviewed a series of approaches to clustering within protein-protein interaction networks. These approaches aim to find functional modules to predict unannotated protein functions based on the structure of an annotated PPI network. However, disparate results can be generated using different approaches and even from the repeated application of a given approach with different parameters. Therefore, these solutions must be carefully compared with predicted results in order to select the approach and parameters which provide the best outcome. Validation is a process of evaluating the performance of the clustering or prediction results derived from different approaches. This section will introduce several basic validation approaches for clustering used in proteomics.

A survey performed by Jiang et al.[42] of clustering of gene expression data revealed three main components to cluster validation: evaluation of performance based on ground truth, an intuitive assessment of cluster quality, and an assessment of the reliability of the cluster sets. These components are also relevant to the evaluation of clustering performance in proteomics.

1.4.1 Validation Based on Agreement with Annotated Protein Function Databases

Clustering results can be compared with ground truth derived from various protein domain databases, such as InterPro, the Structural Classification of Protein (SCOP) database, and the Munich Information Center (MIPS) hierarchical functional categories [13, 21, 48]. These databases are collections of well-characterized proteins that have been expertly classified into families based on their folding patterns and a variety of other information.

In Jiang's et al. [42] work, some simple validation methods are listed which use construction of an $n \times n$ matrix C based on the clustering results, where n is the number of data objects. $C_{ij} = 1$ if object pairs O_i and O_j belong to the same cluster and $C_{ij} = 0$

otherwise. Similarly, a matrix P is built based on the ground truth. Several indices are defined to measure the degree of similarity between C and P.

However, simply counting matches while comparing each predicted cluster against each complex in the data set does not provide a robust evaluation. In cases where each cluster corresponds to a purification, a maximal number of matches will be found, which leads to maximally-redundant results. Krause et al. [48] defined the criteria to assess the fit of the clustering results to the benchmark data set:

1. The number of clusters matching ground truth should be maximal.

2. The number of clusters matching an individual complex should be one.

3. Each cluster should map to one complex only. Clusters matching more than one complex are possibly predicted too inclusive.

4. Complexes should have a similar average size and size distribution to the data set.

Application of these criteria allows a more accurate comparison between clustering results and ground truth, as a one-to-one correspondence is required between predicted clusters and complexes.

1.4.2 Validation Based on the Definition of Clustering

Clustering is defined as the process of grouping data objects into sets by degree of similarity. Clustering results can be validated by computing the homogeneity of predicted clusters or the extent of separation between two predicted clusters. The quality of a cluster C increases with higher homogeneity values within C and lower separation values between C and other clusters.

The homogeneity of clusters may be defined in various ways; all measure the similarity of data objects within cluster C.

$$H_1(C) = \frac{\sum_{O_i,O_j \in C, O_i \neq O_j} Similarity(O_i, O_j)}{\|C\| \cdot (\|C\| - 1)} \tag{1.21}$$

$$H_2(C) = \frac{1}{\|C\|} \sum_{O_i \in C} Similarity(O_i, \overline{O}) \tag{1.22}$$

H_1 represents the homogeneity of cluster C by the average pairwise object similarity within C. H_2 evaluates the homogeneity with respect to the "centroid" of the cluster C, where $\bar{O}$ is the "centroid" of C.

Cluster separation is analogously defined from various perspectives to measure the dissimilarity between two clusters C_1 and C_2. For example:

$$S_1(C_1, C_2) = \frac{\sum_{O_i \in C_1, O_j \in C_2} Similarity(O_i, O_j)}{\|C_1\| \cdot \|C_2\|} \tag{1.23}$$

$$S_2(C_1, C_2) = Similarity(\overline{O}_1, \overline{O}_2) \tag{1.24}$$

1.4.3 Validation Based on the Reliability of Clusters

The performance of clustering results can also be validated by the reliability of clusters, which refers to the likelihood that the cluster structure has not arisen by chance. The significance of the derived clusters is typically measured by the P-value.

In [3], Bu et al. mapped 76 uncharacterized proteins in 48 quasi-cliques in the MIPS hierarchical functional categories. Each protein was assigned a function according to the main function of its hosting quasi-clique. For each cluster, P-values were calculated to measure the statistical significance of functional category enrichment. The P-value is defined as follows:

$$P = 1 - \sum_{i=0}^{k-1} \frac{\begin{pmatrix} C \\ i \end{pmatrix} \begin{pmatrix} G-C \\ n-i \end{pmatrix}}{\begin{pmatrix} G \\ n \end{pmatrix}}, \tag{1.25}$$

where C is the total number of proteins within a functional category and G is the total number of proteins within the graph. The authors regarded as significant those clusters with P-values smaller than $0.01/N_C$ (here N_C is the number of categories).

1.4.4 Validation for Protein Function Prediction

Leave-one-out method

Deng et al. [18] used a leave-one-out method to measure the accuracy of clustering predictions. This method randomly selects a protein with known functions and then hypothetically assumes its functions to be unknown. Prediction methods are then used to predict its functions, and these are compared with the actual functions of the protein. The process is then repeated for K known proteins, $P_i, ..., P_K$. Let n_i be the number of functions for protein P_i in YPD, m_i be the number of predicted functions for protein P_i, and k_i be the overlap between these functions. The specificity (SP) and sensitivity can be defined as

$$SP = \frac{\sum_i^K k_i}{\sum_i^K m_i} \tag{1.26}$$

$$SN = \frac{\sum_i^K k_i}{\sum_i^K n_i} \tag{1.27}$$

Trials using MIPS and other data sets have produced results which are very consistent with those of the distributions of expression correlation coefficients and reliability estimations.

1.5 CONCLUSION

This chapter has provided a review of a set of clustering approaches which have yielded promising results in application to protein-protein interaction networks. Clustering approaches for PPI networks can be broadly differentiated between the classic distance-based methods and the more recently-developed graph-based approaches. Given a network comprised of proteins and their interactions, distance-based clustering approaches assign weights to each protein pair based on their interactions and use classic clustering techniques

to generate predicted clusters. With graph-based approaches, the PPI network is viewed as an unweighted network. Clustering algorithms are employed to identify subgraphs with maximal density or with a minimum cost of cut-off based on the topology of the network. Clustering a PPI network permits a better understanding of its structure and the interrelationship of constituent components. More significantly, it also becomes possible to predict the potential functions of unannotated proteins by comparison with other members of the same cluster.

REFERENCES

1. http://www.plbio.kvl.dk/ dacoj3/resource/yeast_2H.htm.
2. Aebersold, R., Mann, M. Mass spectrometry-based proteomics. *Nature*, 422:198–207, 2003.
3. Albert, R., Barabasi, A. Statistical mechanics of complex networks. *Rev. Mod. Phys*, 74:47–97, 2002.
4. Alon, U. Biological networks: the tinkerer as an engineer. *Science*, 301:1866–1867, 2003.
5. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J. Zhang, Z., Miller, W., Lipman, D.J. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, 25:3389–3402, 1997.
6. Arnau, V., Mars, S., Marin, I. Iterative Cluster Analysis of Protein Interaction Data. *Bioinformatics*, 21:364–378, 2005.
7. Auerbach, D., Thaminy, S., Hottiger, M. O., Stagljar, I. Post-yeast-two hybrid" era of interactive proteomics: facts and perspectives. *Proteomics*, 2:611–623, 2002.
8. Bader, G.D., Hogue, C.W. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4:2, 2003.
9. Barabasi, A.L., Oltvai, Z.N. Network biology: understanding the cell's functional organization. *Nature Reviews*, 5:101–113, 2004.
10. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U. When is "nearest neighbor" meaningful? In *Proceedings of 7th Int. Conf. on Database Theory (ICDT)*, 1999.
11. Blatt, M., Wiseman, S., Domany, E. . Superparamagnetic Clustering of Data. *Phys. Rev. Lett.*, 76:3251–3254, 1996.
12. Blohm, D.H., Guiseppi-Elie, A. . *Curr. Opin. Microbiol.*, 12:41–47, 2001.
13. Bu, D., Zhao, Y., Cai, L., Xue, H., Zhu, X., et al. Topological structure analysis of the protein-protein interaction network in budding yeast. *Nucleic Acids Research*, 31:2443–2450, 2003.
14. Chung, F., Lu, L. The average distances in random graphs with given expected degrees. *Proc. Natl Acad. Sci.*, 99:15879–15882, 2002.
15. Cohen, R., Havlin, S. Scale-free networks are ultra small. *Phys. Rev. Lett.*, 90:058701, 2003.
16. Conrads, T.P., Issaq, H.J., Veenstra, T.D. New tools for quantitative phosphoproteome analysis. *Biochem. Biophys. Res. Commun.*, 290:885–890, 2002.
17. Dandekar, T. et al. Conservation of gene order: a fingerprint of proteins that physically interact. *Trends Biochem. Sci.*, 23:324–328, 1998.
18. Deng, M., Sun, F. and Chen, T. Assessment of the reliability of protein-protein interactions and protein function prediction. *Pac. Symp. Biocomput.*, pages 140–151, 2003.
19. Drees, B. L. Progress and variations in two-hybrid and three-hybrid technologies. *Curr. Opin. Chem. Biol.*, 3:64–70, 1999.

20. Eisen M.B., Spellman P.T., Brown P.O., Botstein D. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci.*, 95:14863–14868, 1998.

21. Enright, A.J., van Dongen, S. and Ouzounis, C.A. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, 30:1575–1584, 2002.

22. Fell, D.A., Wagner, A. The small world of metabolism. *Nat. Biotechnol.*, 18:1121–1122, 2000.

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

24. Fransen, M., brees, C., Ghys, K., Amery, L. et al. aa. *Mol. Cell Proteomics*, 2:611–623, 2002.

25. Gavin, A.C. et al. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, 415:141–147, 2002.

26. Ge, H. . *Nucleic Acids Res.*, 28:1–7, 2000.

27. Getz, G., Levine, E., Domany, E. Coupled two-way clustering analysis of gene microarray data. *Proc. Natl. Acad. Sci.*, 97:12079–12084, 2000.

28. Getz, G., Vendruscolo, M., Sachs, D., Domany, E. Automated assignment of SCOP and CATH protein structure classifications from FSSP scores. *Proteins*, 46:405–415, 2002.

29. Giot, L. et al. A protein interaction map of Drosophila melanogaster. *Science*, 302:1727–1736, 2003.

30. Glazko G., Gordon A., Mushegian A. The choice of optimal distance measure in genome-wide data sets. , :, 2005.

31. Glover,F. . Tabu search. *ORSA J. Comput.*, 1:190–206, 1989.

32. Goldberg,D.S., Roth,F.P. Assessing experimentally derived interactions in a small world. *Proc. Natl acad. Sci.*, 100:4372–4376, 2003.

33. Hartuv, E., Shamir, R. A Clustering Algorithm based Graph Connectivity. *Information Processing Letters*, 76:175–181, 2000.

34. Hartwell, L.H., Hopfield, J.J., Leibler, S., Murray, A.W. From molecular to modular cell biology. *Nature*, 402:C47–C52, 1999.

35. Ho, Y. et al. Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature*, 415:180–183, 2002.

36. Ito, T., Chiba, T., Ozawa, R., Yoshida, M. et al. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci.*, 98:4569–4574, 2001.

37. Ito, T., Ota, K., Kubota, H., Yamaguchi, Y., Chiba, T., Sakuraba, K., Yoshida, M. Roles for the two-hybrid system in exploration of the yeast protein interactome. *Mol. Cell Proteomics*, 1:561–566, 2002.

38. Jain A., Murty M., Flynn P. Data clustering: a review. *ACM Computing Surveys*, 31:264–323, 1999.

39. Jansen, R. A. et al. A Bayesian Networks Approach for Predicting Protein-Protein Interactions from Genomic Data. *Science*, 302:449–453, 2003.

40. Jansen, R. et al. Relating Whole-Genome Expression Data with Protein-Protein Interactions. *Genome Research*, 12:37–46, 2002.

41. Jeong, H., Mason, S.P., Barabási, A.-L., Oltvai, Z.N. Lethality and centrality in protein networks. *Nature*, 411:41–42, 2001.

42. Jiang, D., Tang, C., Zhang, A. Cluster Analysis for Gene Expression Data: A Survey. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 16:1370–1386, 2004.

43. Johnsson, N., Varshavsky, A. Split Ubiquitin as a Sensor of Protein Interactions In vivo. *Proc. Natl. Acad. Sci.*, 91:10340–10344, 1994.

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

45. King, A. D., Przulj, N., Jurisica, I. Protein complex prediction via cost-based clustering. *Bioinformatics*, 20:3013–3020, 2004.

46. Koonin, E.V., Wolf, Y.I., Karev, G.P. The structure of the rotein universe and genome evolution. *Nature*, 420:218–223, 2002.

47. Korbel J.O., Snel B., Huynen M.A., Bork P. SHOT: a web server for the construction of genome phylogenies. *Trends Genet*, 18:159–162, 2002.

48. Krause, R., von Mering, C. and Bork, P. A comprehensive set of protein complexes in yeast: mining large scale protein-protein interaction screens. *Bioinformatics*, 19:1901–1908, 2003.

49. Kumar A., Snyder M. Protein complexes take the bait. *Nature*, 415:123–124, 2002.

50. Kuster, B., Mortensen, P., Andersen, J.S., Mann, M. Mass spectrometry allows direct identification of proteins in large genomes. *Protemics*, 1:641–650, 2001.

51. Lasonder, E. et al. Analysis of the Plasmodium falciparum proteome by high-accuracy mass spectrometry. *Nature* , 419:537–542, 2002.

52. Lebowitz, J., Lewis, M.S., Schuck, P. Modern analytical ultracentrifugation in protein science: A tutorial review. *Protein Sci.*, 11:2067–2079, 2002.

53. Li, S. et al. A map of the interactome network of the metazoan. *Science*, 303:540–543, 2004.

54. MacBeath G., Schreiber, S.L. Printing Proteins as Microarrays for High-Throughput Function Determination. *Science*, 289:1760–1763, 2000.

55. Mann, M. et al. Analysis of protein phosphorylation using mass spectrometry: deciphering the phosphoproteome. *rends Biotechnol*, 20:261–268, 2002.

56. Mann, M., Jensen, O.N. Proteomic analysis of post-translational modifications. *Nature Biotechnol*, 21:255–261, 2003.

57. Marcotte, E. M. et al. Detecting Protein Function and Protein-Protein Interactions from Genome Sequences. *Science*, 285:751–753, 1999.

58. Marcotte, E. M. et al. Detecting Protein Function and Protein-Protein Interactions from Genome Sequences. *Nature*, 402:83–86, 1999.

59. Maslov, S., Sneppen, K. Specificity and stability in topology of protein networks. *Science*, 296:910–913, 2002.

60. Mewes, H. W. at al. MIPS: analysis and annotation of proteins from whole genomes. *Nucleic Acids Research*, 32:D41–D44, 2004.

61. Milgram, S. The small world problem. *Psychol. Today*, 2:60, 1967.

62. Mirkin B., Koonin E.V. A top-down method for building genome classification trees with linear binary hierarchies. *Bioconsensus*, 61:97–112, 2003.

63. Newman, M. E. Network construction and fundamental results. *Proc. Natl. Acad. Sci.*, 98:404–409, 2001.

64. Nooren, I.M.A., Thornton, J.M. Diversity of protein-protein interactions. *EMBO J.*, 22:3486–3492, 2003.

65. Ofran, Y., Rost, B. Analyzing six types of protein-protein interfaces. *J. Mol. Biol.*, 325:377–387, 2003.

66. Otzen, D.E., Fersht, A.R. Analysis of protein-protein interactions by mutagenesis: direct versus indirect effects. *Protein Eng.*, 12:41–45, 1999.

67. Oyama, T. et al. Extraction of knowledge on protein-protein interaction by association rule discovery. *Bioinformatics*, 18:705–714, 2002.

68. Patterson, S.D., Aebersold, R.H. Proteomics: the first decade and beyond. *Nature Genetics*, 33:311–323, 2003.

69. Pei, P. and Zhang, A. A topological measurement for weighted protein interaction network. In *Proceedings of the IEEE Computer Society Bioinformatics Conference (CSB 05)*, pages 268–278, 2005.

70. Pei, P. and Zhang, A. A two-step approach for clustering proteins based on protein interaction profile. In *Fifth IEEE International Symposium on Bioinformatic and Bioengineering (BIBE 2005)*, pages 201–209, 2005.

71. Pellegrini, M. et al. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *PNAS*, 96:4285–4288, 1999.

72. Peng, J., Elias, J.E., Thoreen, C.C., Licklider, L.J., Gygi, S.P. Evaluation of multidimensional chromatography coupled with tandem mass spectrometry (LC/LC-MS/MS) for large-scale protein analysis: the yeast proteome. *J. Proteome res.*, 10:1021, 2002.

73. Pereira-Leal, J.B., Enright, A.J., Ouzounis, C.A. Detection of functional modules from protein interaction networks. *Proteins: Structure, Function, and Bioinformatics*, 54:49–57, 2004.

74. Phizicky, E.M., Fields, S. Protein-protein interactions: methods for detection and analysis. *Microbiol. Rev.*, 59:94–123, 1995.

75. Rives, A.W. and Galitski, T. Modular organization of cellular networks. *Proc. Natl. Acad. Sci.*, 100(3):1128–33, 2003.

76. Samanta, M.P. and Liang,S. Redundancies in large-scale protein interaction networks. *Proc. Natl Acad. Sci.*, 100:12579–12583, 2003.

77. Sigman, M., Cecchi, G. A. Global organization of the Wordnet lexicon. *Proc. Natl. Acad. Sci.*, 99:1742–1747, 2002.

78. Sole, R.V., Pastor-Satorras, R., Smith, E., Kepler, T.B. A model of large-scale proteome evolution. *Adv. Complex Systems*, 5:43–54, 2002.

79. Spinozzi, F., Gazzillo, D. Giacometti, A., Mariani, P., Carsughi, F. Interaction of proteins in solution from small angle scattering: a perturbative approach. *J. Biophys*, 82:2165–2175, 2002.

80. Spirin, V. and Mirny, L.A. Protein complexes and functional modules in molecular networks. *Proc. Natl. Acad. Sci.*, 100:12123–12128, 2003.

81. Tavazoie, S., Hughes, D., Campbell, M.J., Cho, R.J., Church, G.M. Systematic determination of genetic network architecture. *Nature Genet*, pages 281–185, 1999.

82. Van Dongen, S. A new cluster algorithm for graphs. Technical Report INS-R0010, Center for Mathematics and Computer Science (CWI), Amsterdam, 2000.

83. Van Dongen, S. Performance criteria for graph clustering and markov cluster experiments. Technical Report INS-R0012, Center for Mathematics and Computer Science (CWI), Amsterdam, 2000.

84. Veselovsky, A.V., Ivanov, Y.D., Ivanov, A.S., Archakov, A.I.J. Protein-protein interactions: mechanisms and modification by drugs. *Mol. Recognit*, 15:405–422, 2002.

85. Vidal, M. *The Two-Hybrid System*, page 109. Oxford University Press, 1997.

86. von Mering, C., Krause, R., Snel, B., Cornell, M., Oliver, S.G. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417:399–403, 2002.

87. Wagner, A. The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes. *Mol. Biol. Evol.*, 18:1283–1292, 2001.

88. Wagner, A. How the global structure of protein interaction networks evolves. *Proc. R. Soc. Lond.*, 270:457–466, 2003.

89. Washburn, M.P., Wolters, D., Yates, J.R. Large-scale analysis of the yeast proteome by multidimensional protein identification technology. *Nature Biotechnol*, 19:242–247, 2001.

90. Watts, D. J. *Small worlds*, page . Princeton University Press, 1999.

91. Yanagida, M. Functional proteomics; current achievements. *J. Chromatogr. B*, 771:89–106, 2002.

92. Zhang, B., Kraemer, B., SenGupta, S., Fields, S., Wickens, M. Yeast three-hybrid system to detect and analyze interactions between RNA and protein. *Methods Enzymol*, 306:93–113, 1999.

93. Zhou, H. Distance, dissimilarity index, and network community structure. *Physical Review*, E67:061901, 2003.

94. Zhou, H. Network landscape from a Brownian particle's perspective. *Physical Review*, E67:041908, 2003.

95. Zhu, H., Bilgin, M., Bangham, R., Hall, D. et al. . *Science*, 293:2101–2105, 2001.

96. Zhu, H., Bilgin, M., Snyder, M. Proteomics. *Annual Review of Biochemistry*, 72:783–812, 2003.